Design and Gait Synthesis for a 3D Lower Body Humanoid

by

Safwan Choudhury

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Bipedal locomotion is a challenging control engineering problem due to the non-linear dynamics and postural instability of the bipedal form. In addition to these challenges, some dynamical effects such as the ground reaction force are difficult to model accurately in simulation. To this end, it is essential to develop physical hardware to validate walking control strategies and gait generation methods. This thesis develops an on-line walking control strategy for humanoid robots and the electromechanical design of a physical platform for experimental validation.

The first part of the thesis presents the development of a 14 degrees-of-freedom (DOF) lower body humanoid robot. The initial electromechanical design of the proposed system is derived from dynamic modeling of a general multibody system. Kinematic trajectories for the lower body joints are extracted from motion captured human gait data to form the preliminary design specifications. The drivetrain components are selected by analyzing the mechanical power requirements, torque-speed profiles, efficiency and thermal characteristics of actuators. The supporting mechanical chassis and power transmission system are designed to raise the center-of-mass (to reduce the swinging inertia of each leg) while minimizing the overall weight of the system.

Refining the design of a complex multibody robotic system like the biped is an iterative process. The mechanical model of the system is transferred from Computer-Aided-Design (CAD) software to a dynamic simulator for analysis and the design is revised to improve performance. This iterative approach is necessary as small changes in the mechanical model can have significant impact on the overall dynamics of the system as well as implications for control design. A streamlined prototyping toolchain is developed in this thesis to extract the relevant kinematic/dynamic parameters of a mechanical system in CAD and automatically generate the equivalent system in a dynamic simulator. This toolchain is used to revise the electromechanical design and generate forward dynamics simulations.

The second portion of this thesis develops a novel walking control strategy for on-line gait synthesis for 3D bipedal robots based on Wight's Foot Placement Estimator (FPE) algorithm. This algorithm is used to determine the desired swing foot position on the ground to *restore* balance for a 2D bipedal robot. The FPE algorithm is extended to the general 3D case by selecting a suitable plane in the desired direction of motion. Complete gait cycles are formed by combining a finite state machine with the 2D FPE solution along the selected plane. Gait initiation is accomplished by computing state-dependent task space trajectories on-line to produce a forward momentum along the selected plane. A whole-body motion control framework (Jacobian-based prioritized task space control scheme)

tracks the task space trajectories and generates the appropriate joint level command for each state. The joint level commands are tracked by local high gain PD controllers. This framework produces the desired whole-body motion during each state while satisfying higher priority constraints. Gait termination is accomplished by controlling the swing foot position to track the FPE point on the ground along the selected plane.

The proposed control strategy is verified in simulation and experiments. A parallel hardware-in-the-loop (HIL) testing environment is developed for the physical lower body humanoid robot. The motion control framework and joint dynamics used in the proposed walking control strategy are verified through HIL experiments.

Acknowledgements

I would like to thank my supervisor Dana Kulić for her invaluable guidance and support throughout my research. I'm extremely grateful for her advice, paper editing, debugging sessions and countless meetings over the last two years.

I would also like to thank the following people for their assistance during the course of my masters research:

- Derek Wight for his advice on electromechanical design, developing models with QUARC and help with the HIL experiments. I am also grateful for the opportunity to work at Quanser Inc under the NSERC Industrial Postgraduate Scholarship. The experience was extremely valuable to the development of our streamlined toolchain during the design process.

- Christopher Kirby from Christie Digital Systems for taking the time to to provide guidance on the mechanical design. As an electrical engineering undergraduate with minimal experience and knowledge of mechanical engineering, his advice on best practises, designing the mechanical power transmission system for the biped and generating CAD drawings was invaluable.

- Matt Millard of the Real Motion Group in Systems Design Engineering for his expertise in the field of gait analysis and for help with a wide range of topics pertaining to walking robots and contact modeling.

- Charles Boyle at the Engineering Machine Shop on campus for his advice and assistance with manufacturing the bipedal robot. Charlie often stayed after hours and helped with machining various parts of the project.

Lastly, I would like to thank my parents, sister and my wife for their endless love, support and encouragement throughout my academic career at the University of Waterloo.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Bipedal locomotion is a necessary capability to enable humanoid robots to navigate a wide range of environments required for most practical real-world applications. However, bipedal locomotion is a very challenging control problem due to the postural instability of the bipedal form. The act of bipedal walking is a cyclical repetition of a single gait cycle, a pattern of limb movement, which consists of a stance phase (foot is making contact with the ground) and a swing phase (foot is off the ground reaching for the next stance position). Bipedal gait can also be separated into the single and double support phases, when one or both feet are in contact with the ground, respectively.

The most popular technique to achieve walking for bipedal robots thus far has been through control strategies based on the Zero-Moment Point (ZMP) criterion [1]. ZMP-stable trajectories are often computed off-line. Typically, these strategies [2, 3] are energetically inefficient since they are actively trying to maintain balance by keeping the ZMP within the region of foot support. Furthermore, the resulting gait does not utilize the natural dynamics of the system and consequently does not look "human-like".

A parallel research direction has considered taking advantage of the structure dynamics to achieve a stable gait cycle without active control effort. This pioneering research by McGeer [4] introduced a unique class of legged robots known as passive dynamic walkers [5]. These robots are designed to walk on an inclined surface so that the passive mechanism is powered by gravity alone [6]. In addition to producing highly efficient walk, the gait patterns generated using this approach are much more human-like in comparison to ZMP-based control. However, passive dynamic walkers lack robustness to perturbations due to very narrow regions of attraction.

In this thesis, a novel walking control strategy which has the potential to increase robust-

ness and energetic efficiency is presented. To enable experimental validation with physical hardware, a 14 degree-of-freedom (DOF) lower body humanoid robot was developed. The ultimate goal is to validate the effectiveness and usefulness of the presented walking control strategy by testing it on the physical biped hardware.

At the core of the proposed walking control strategy is the Foot Placement Estimator (FPE) algorithm which was previously developed for 2D bipedal robots [7]. The FPE algorithm determines where an unstable biped must step to restore balance. One of the key novelties of the FPE theory is the wide range of applicability of the results. It can be used to augment existing walking control strategies or be extended to form dynamically stable gait cycles [8].

The solution to the FPE equation itself can be used as a recovery mechanism (i.e. in the face of a destabilizing disturbance) with existing ZMP-based strategies. Alternatively, it can be used to increase the narrow regions of attraction which plague minimally actuated passive dynamic walkers [9, 10, 11].

1.1 Contributions

1. **Electromechanical Design of 14 DOF Bipedal Robot**
   The development of a 14 DOF bipedal robot for the purposes of research in bipedal locomotion is presented. Drivetrain component selection is performed using gait estimation and accounting for actuator dynamics. The mechanical chassis is designed with anthropometric dimensioning and the weight distribution is manipulated to make the physical system easier to control.

2. **Rapid Prototyping Toolchain Development**
   A toolchain is developed to streamline the iterative design process by automatically generating dynamic simulations with 3D visualization directly from Computer-Aided-Design (CAD) software. It enables fast incremental changes to revise the mechanical design and immediately analyze the resulting behaviour in simulation. The toolchain is used to improve the initial design and compute forward dynamics for the 14 DOF bipedal robot.

3. **Extension of FPE Algorithm to 3D**
   The 2D FPE algorithm is extended to form complete gait cycles for a 3D bipedal robot. The proposed algorithm selects a 2D plane in the chosen direction of motion and generates trajectories to produce a forward moving momentum along the

plane. A whole body motion control framework coupled with a finite state machine is used to track the generated trajectories and form dynamically stable gait cycles. The efficacy of the proposed walking control strategy is demonstrated through side stepping and forward walking in dynamic simulations.

4. **Experimental Validation on Physical Hardware**
   Dynamic simulation models are designed with a shared code base, allowing the same controller to target either a simulated plant or the constructed 14 DOF biped using hardware-in-the-loop (HIL). The validity of the motion control framework and actuator dynamics models are demonstrated on the physical hardware using HIL.

1.2 Outline

The remainder of this thesis is organized as follows. Chapter 2 outlines existing walking control strategies in the robotics literature. While most control strategies use a dynamic measure of balance to form complete gait cycles, some newer approaches are similar to the strategy presented in this thesis. These control strategies aim to use foot placement as a means to form robust and energetically efficient gait for bipedal locomotion.

The initial electromechanical design and development of a 14 DOF lower body humanoid robot (biped) is presented in Chapter 3. Human gait trajectories are used as an approximate starting point for actuators and drivetrain component selection. While it is understood that using human gait as a reference is ambitious, this approach results in a conservative initial design by selecting components capable of human-speed gait, which is significantly faster than is achievable by most current humanoid designs. The forward dynamics and 3D visualization for the final prototype design is generated using the toolchain presented in Chapter 4. This is used to test control strategies for the biped in simulation prior to implementing it on the physical hardware.

One of the key challenges in developing physical hardware for bipedal locomotion research is the impact of small design changes on the overall stability of the system. A toolchain is developed in Chapter 4 to help improve the initial design and ultimately produce a system with adequate drivetrain performance. The toolchain streamlines the process of designing a mechanical prototype in Computer-Aided-Design (CAD) software and immediately analyzing its behaviour in dynamic simulations.

The extension of the 2D FPE theory to the 3D case is presented in Chapter 5. A motion control framework is presented in order to simultaneously use the FPE algorithm to achieve

walking and maintain stability throughout the gait cycle. The 2D FPE theory and the proposed extension to 3D are verified through dynamic simulations.

Experiments validating the actuator dynamics and motion control framework on the physical hardware are described in Chapter 6. The actuator dynamics models are verified to match the physical DC motors through HIL simulations. The motion control framework used for extending the FPE theory to 3D is also verified on physical hardware using HIL.

Lastly, conclusions and future work regarding the proposed walking control strategy and developed physical hardware are presented in Chapter 7.

Chapter 2

Related Work

Robotic systems capable of locomotion have existed for decades, with some of the earliest designs dating back to the late nineteenth century. Using a series of mechanical drivetrain components, Rygg developed a mechanical horse capable of generating fixed gait in 1893 [12]. Several years earlier in 1888, Fallis developed the first *passive* bipedal walking toy capable of stable gait [13]. The term *passive* denotes the lack of any active power input to the system. Fallis's toy produced *statically* stable walking, whereby the center-of-mass (COM) always remains within the base of support throughout the gait cycle. In contrast, the COM leaves the base of support during *dynamically* stable gait.

McGeer's seminal research [4] nearly a century later produced *passive* bipeds capable of dynamically stable gait without any control effort. This work led to a new class of passivity-based designs for bipedal robots. Starting in the late 60's, pivotal research by Vukobratovic [14] introduced the Zero-Moment Point (ZMP), a dynamic measure of balance widely used in walking control strategies today. This sparked the rapid growth of actively-powered bipedal robot designs. The electromechanical design and gait synthesis method used to achieve bipedal locomotion are closely related. Bipedal robots are often designed to target a specific control strategy. This chapter presents a literature review of the related work in this research area.

2.1 Humanoid Electromechanical Design

The first *active* bipedal robot, WL-5 was developed in the 1972 by Kato et al. at Waseda University [15]. Since then, the number of bipedal robots being developed for commercial

and research applications has been growing rapidly. The main electromechanical design considerations for humanoid robots are reviewed in this section.

2.1.1 Degrees of Freedom

A bipedal robot achieves locomotion by articulating the limbs connected by joints in the lower body. Therefore, the number and configuration of the degrees-of-freedom (DOF) in the lower body of the robot play an important role in bipedal locomotion. The human body is an obvious design inspiration for bipedal systems. While a few designs have aimed to model the large number of DOF formed by muscles and tendons [16, 17], the more common approach has been to select a reduced number of DOF, modeled on the skeletal system, to achieve walking.

The primary DOF used to articulate the lower body limbs are the hip, knee and ankle joints. A simpler design widely used in robotics research is the 2-dimensional (2D) planar biped. This approach typically uses a 1 DOF hip and 1 DOF knee joint in conjunction with point feet [18]. Physical implementations of these planar bipeds constrain its motion to the sagittal plane by using a boom [19, 20, 7].

The most common examples of *active* 3-dimensional (3D) lower body designs use 3 DOF for the hip joint, 1 DOF at the knee and 2/3 DOF for the ankle joint [21, 22, 23, 24, 25]. The hip joint is arguably the most complex joint to design as it must provide a wide range of motion and be capable of carrying the weight of each leg. Several bipedal robots use innovative hip joint designs to improve mobility and/or reduce the DOF. Aldebaraan's Nao robot strategically places the hip motors to eliminate 1 DOF while maintaining a similar range of motion [26]. The bipedal robot from the AIST's Humanoid Robot Project, HRP-2 utilizes a unique cantilever hip structure to enable cross-legged walking (e.g. on a tight rope) [23].

More recent research has been aimed at mimicking the large number of DOF present in humans through complex electromechanical designs. Musculoskeletal robots [16, 17] generate a high (variable) number of DOF using complex actuation mechanisms to emulate muscles. The additional complexity and large DOF drastically increase the complexity of the control strategy.

2.1.2 Mechanism Design

The mechanical design of linkages used in robotics largely fall into two categories, serial and parallel link manipulators. Serial link manipulators contain a series of joint-driven links

that extend from the base to an end-effector. Parallel link manipulators on the other hand contain several (serial) links connecting a base to its end-effector. A well-known parallel link manipulator is the 6 DOF Stewart-Gough platform [27, 28]. The same parallel design technique can be adapted to design high performance multi-DOF joints [29]. A common application of this design approach in the lower body of a humanoid is for a spherical (i.e. 3 DOF) hip joint [30].

The primary motivation for investigating serial vs. parallel mechanism design for lower body humanoid robots is to improve the performance of the system. The majority of the mass of a robot joint-link assembly comes from the actuator. It is often useful to manipulate the overall weight distribution to improve the system behaviour under control. For example, a common approach is to approximate the dynamics during the single support phase as an inverted pendulum [31]. In this approach, the swinging leg is assumed to be massless. Having larger motors further down along the swing leg changes the inertia and location of the COM, consequently influencing the stability during walking [32]. Therefore, it is common to manipulate the weight distribution in the lower body to raise the COM and improve the system performance.

Serial link manipulators have the benefit of simpler design and kinematics, making it easier to identify singular configurations in the joint space. Some of the most famous humanoid robots use serial mechanisms, including the Honda ASIMO [33] and Waseda University's WABIAN-2 [24]. The serial arrangement of actuator-driven joints also makes it difficult to relocate actuators from each joint axis for better weight distribution. Mechanical transmission components (e.g. belts, pulleys, gears) are often used to decouple the actuator output from the joints [24]. However, this also introduces additional complexity which is difficult to model (i.e. drive flexibility introduced by a belt-pulley system) and often requires tuning [22].

On the other hand, the parallel link manipulators [34] enable easier relocation of actuators. Parallel joints also distribute the torque requirements equally across all actuators, which reduces the individual actuator size and improves the power-to-weight ratio [35]. Due to these reasons, it was found that the energy consumption of parallel linked structures is significantly lower than that of its serially linked counter part [32]. Parallel joints also provide the benefit of high stiffness and accuracy [36]. However, there are several drawbacks to designing parallel link manipulators. It is more difficult to determine the singular configurations of the parallel structure without kinematic analysis [36]. They also increase the design complexity and have a reduced range of motion, making the realization more difficult for 3D bipedal robots [37]. Parallel mechanisms also suffer from greater friction (due to a larger number of interacting components), which is difficult to model and compensate [38, 39].

2.1.3 Passive Walkers

The most simple bipedal robot designs are purely mechanical devices which are capable of walking without any active power, relying on the natural dynamics of the system for gait generation. McGeer first demonstrated this concept on a simple mechanical system consisting of two straight legs connected by a hinge at the hip [4], introducing a class of bipedal robots known as "passive walkers". This initial design was revised by adding knees for more anthropomorphic gait and to solve the foot clearance issue during the recovery phase [40]. Nearly a decade later, Garcia et al. performed an in-depth analysis of the speed, efficiency and mechanism design of 2D passive dynamic walkers [41]. Several design options were investigated for passive 2D bipeds, including kneed and straight-legged models with round and point feet. A four legged lower body design (inner and outer pairs) was adopted for physical implementations to constrain the problem to 2D and prevent falling sideways. Collins et al. built the first 3D passive bipedal robot which used swinging arms to counteract the sideways instability [42]. This passive design was capable of producing remarkably human-like gait without any active power input.

2.1.4 Active Walkers

The purely passive mechanical designs relied on gravitational power and sloped inclines. To achieve walking on level ground, actuation is required. Most active walkers use electric motors, pneumatics or hydraulics as actuators. Both fully and partially actuated designs have been proposed to achieve locomotion.

Actuation Type

Electric motors are most commonly used to power active walkers. Electromechanical designs based on this type of actuation use AC motors, DC motors or servomotors. Some humanoid robots have custom built electric motors designed specifically to improve the performance for the intended application (e.g. the Honda humanoid robots [33, 21]). Electric motors drive the link motion through either direct or geared drive mechanisms. In direct drive, the motor shaft is mounted to drive the joint directly producing low torques and high speed. Geared drives are often used to increase the motor torques and lower the shaft speed through a gear reduction ratio. Standard gearing mechanisms (e.g. planetary, spur gears) suffer from the backlash problem while other high end (expensive) options like harmonic drives do not. Waseda University's WABIAN-2 robot combines a DC motor with harmonic drive [24]. Alternatively, KAIST's HUBO robot selectively uses harmonic drives only in

the lower body since backlash in these joints can affect the overall stability of the system [22]. The major drawback of using electric motors is their high mechanical impedance (no compliance) and drivetrain losses when coupled with geared drive mechanisms.

Pneumatics is used to generate mechanical power through pressure manipulation within a tube by an air compressor. The main advantage of this actuation strategy is its high power-to-weight ratio and inherent compliance [43]. Electromechanical designs based on this type of actuation use variations of Pneumatic Artificial Muscles (PAM) to control joint motion. The most common variation used in lower body joints is the McKibben muscle (braided PAM). Delft University has produced several robots (Mike [44], Max [45] and Denise [46, 47]) which use McKibben muscles. VUB's planar biped Lucy based on pleated PAM actuators was used to explore energetically efficient walking with adaptable compliance [48]. The major drawback of pneumatic actuators is their low position accuracy and non-linear response.

Hydraulics is used to generate mechanical power through pressure manipulation of oil from a hydraulic pump. Flow control valves are used to regulate the hydraulic pressure in a hollow cylinder, which in turn manipulates a linear force applied through a piston. The linear force is applied to a moment arm producing a torque at the joint. The main advantage of this actuation strategy is its high power density and inherent compliance. The non-linear properties of hydraulic actuators make joint level control a challenging task. Impedance control schemes are used to reduce the non-linear effects and compensate for unmodeled dynamics [49]. In addition to the modeling complexities, the major drawbacks of using hydraulic actuators are the cost and the pump required to supply the high pressure oil through hydraulic hoses to each joint. This increases the design complexity and introduces additional points of failure.

Actuation Strategy

The actuation strategy used for most powered walkers has relied on complete (fully active) control of every joint. Electric motors are most commonly selected for bipedal robot designs using this strategy. However, some designs like the force-controllable SARCOS humanoid CB use hydraulic actuators [50].

A more recent actuation strategy extends passive dynamics principles to produce energetically efficient active walkers. To achieve walking on level ground, the passivity-based design techniques were extended by substituting gravitational power with minimal actuation [6, 5]. This produced energetically efficient 3D active walkers. Most minimally actuated walkers

have electromechanical designs aimed towards obtaining high compliance and low control stiffness.

Delft University's pneumatic biped Denise was designed with cylindrically shaped feet and a mechanical link to couple the forward motion of one leg to the reverse motion of the other [51]. Minimal actuation is provided at the hip joint to achieve walking on level ground. Flame, the most recent energetically efficient 3D bipedal robot from Delft University, was designed specifically to take advantage of the natural dynamics of the system [46] using series elastic actuators (SEA) for compliance [52].

Another approach to designing minimally actuated walkers is to use actuation at the ankle joints [5, 53, 54, 55], often in addition to the hip joint. Meta, another energetically efficient walker from Delft University, uses two motors at the ankle in addition to the two motors at the hip joint [54]. Ankle actuation has been shown to improve the versatility of some walkers by enabling variable walking speeds [55]. When both hip and ankle actuation is used, the key difference from fully actuated systems to the minimally actuated ones is that the joints are not actuated through the full gait cycle, but only at certain key phases (e.g. push off).

2.2 Walking Control Strategies and Gait Generation

There exists a wide range of walking control strategies and gait generation methods to achieve bipedal locomotion. Some of the most popular strategies are briefly reviewed in this section.

2.2.1 Zero-Moment Point

The most popular techniques to achieve walking have been trajectory generation and control strategies based on the Zero-Moment Point (ZMP) criterion [1]. The ZMP defines a point on the ground where the forces acting on a biped do not produce a moment about the axes parallel to the ground plane (shown in Figure 2.1). The criterion states that the biped is stable if the ZMP is kept within the region of foot support at every time instant. The ZMP location is typically used as a feedback mechanism to achieve dynamically stable gait.

This criterion can also be used to pre-compute ZMP-stable trajectories off-line. Huang et al. presented a walking pattern generation algorithm to compute the desired foot and

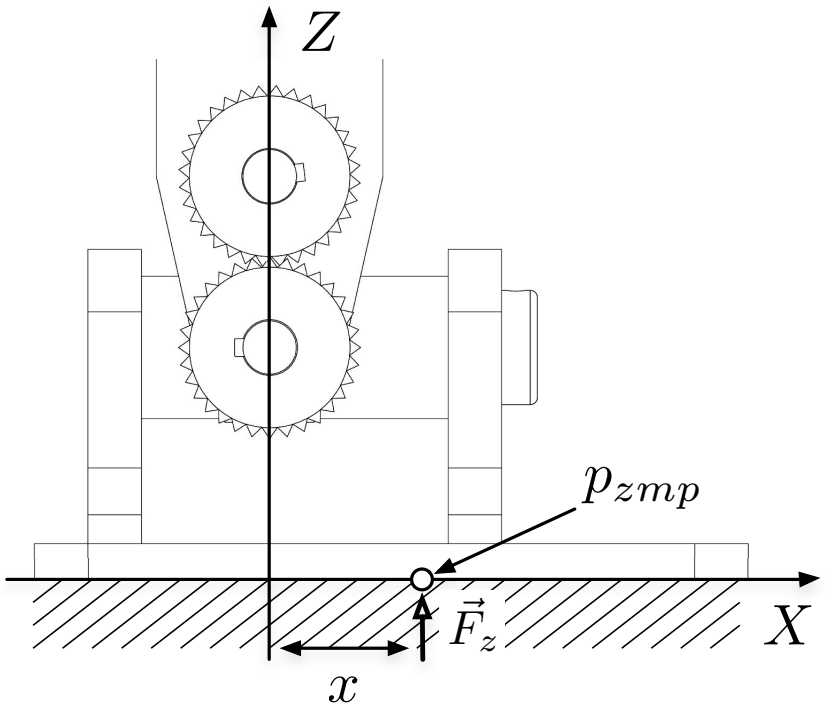


Figure 2.1: Planar view of the ZMP location (p_{zmp}) within the region of foot support.

hip trajectories off-line [56]. This algorithm aimed to select a ZMP-stable hip trajectory which maximized the stability margin (distance from the ZMP point to the boundary of the support region). Computationally expensive exhaustive search was used off-line to obtain the optimal solution for a given walking speed and step length. During the on-line phase, the selected optimal trajectories were tracked to execute walking.

Kajita et al. developed a walking pattern generation algorithm based on the 3D linear inverted pendulum mode (LIPM) [57] and preview control [58]. Using this approach, the desired swing foot positions can be adjusted on-line to compute a new ZMP trajectory which remains within the region of foot support. A ZMP tracking controller (based on the table-cart model) is used to generate the desired *future* COM trajectory. The preview controller uses this future reference signal (within a specified time period) to generate the adjusted *current* COM trajectory to remain dynamically stable. More recent methods use more accurate modeling techniques (instead of 3D LIPM) to compute ZMP-based trajectories on-line [59].

The biggest drawback of ZMP-based methods is that the resulting trajectory does not provide any strategy to respond to disturbances due to uneven terrain or unexpected forces. Typically, these strategies are also energetically inefficient since they are actively trying to maintain balance by keeping the ZMP point within the region of foot support. Furthermore, the resulting gait does not utilize the natural dynamics of the system and consequently does not look human-like.

Despite the drawbacks, some of the most popular humanoid robots for commercial and research applications utilize some variation of ZMP based feedback for bipedal locomotion. The well known Honda ASIMO [33] uses ZMP as part of its i-WALK technology (predicted movement control). Newer bipeds like Aldebaraan's Nao [60] robot contain a pre-packaged walking algorithm which implements ZMP-based walking pattern generation [61].

2.2.2 Virtual Passive Dynamics

The preliminary research in the field of passive dynamics was restricted to the specific scenario of walking on an incline (using gravitational power). This approach lacked the versatility that is required for humanoid robots if they are to ultimately navigate in most human environments. These walkers also lacked robustness to perturbations due to very narrow regions of attraction. Further investigative research was helpful in characterizing the nature of a passive gait cycle [9] in terms of stability and energy usage. This ultimately gave rise to new strategies which were aimed towards emulating the work done by gravity on an inclined slope [10]. These hybrid approaches (known as virtual passive dynamics) improved the versatility of the simple legged robots to walk on level ground with minimal actuation as a replacement for gravity [62].

While these minimally actuated bipeds have been shown to exhibit similar energetics as purely passive machines [63], most of the initial research was restricted to the 2D dynamics in the sagittal plane. The key challenge of extending these simple models to 3D (i.e. incorporating lateral dynamics) was unstable motion introduced by mismatch of the roll velocity and contact conditions [11]. This was particularly challenging since small disturbances were able to completely destabilize a passive dynamic walker (due to the narrow region of attraction). These challenges were addressed in part by tweaking the mechanical design for better compliance and introducing minimal actuation (as discussed in Section 2.1.4). Full 3D bipedal robots capable of energetically efficient walking with minimal actuation have been realized [5, 51, 46]. However, these systems still share the same underlying weakness to large perturbations.

2.2.3 Force/Impedance

Pratt developed a motion control framework that uses abstract virtual components to generate compliant joint torques [64]. This approach aims to augment the natural dynamics of the system by emulating the effects of virtual components (e.g. springs, dampers, dashpots). The mapping between the virtual forces and emulated joint torques are obtained through the kinematic model [65]. Pratt demonstrated walking with this approach on two bipedal robots with series elastic actuators (SEA), the Spring Flamingo and Spring Turkey [66]. Using this motion control framework improves the biped's robustness to rough terrain and unexpected disturbances.

The Dynamic Balance Force Control (DBFC) algorithm developed by Stephens et al. developed a method for tracking desired COM motion for compliant humanoid robots [67]. This approach uses Pratt's virtual model control framework (DBFC-VMC) to achieve generic force controlled tasks. The DBFC-VMC controller is extended beyond balancing to form complete gait cycles using preview control [58] based on 3D LIPM dynamics to track COM trajectories for walking.

A similar robust and compliant control strategy capable of handling rough terrain is presented by Hyon et al. [68]. This approach uses a passivity based force-controlled framework to balance and redistribute the ground reaction force over the supporting contact points. This framework was demonstrated on the hydraulically actuated SARCOS humanoid. Ott et al. developed a similar compliant balancing and posture controller for the DLR biped [69].

2.2.4 Machine Learning

Another popular method of achieving bipedal locomotion is through machine learning approaches including neural networks, central pattern generators (CPG) and other oscillators.

The types of neural networks used for gait synthesis vary from multilayer perceptrons, recurrent neural networks to Cerebellar Model Arithmetic Controllers (CMAC) [70]. A notable development in this space is Miller's hierarchical control strategy which combines simple gait oscillators and learning through CMAC neural networks [71, 72]. This approach does not require detailed information of the system dynamics to achieve walking. The learning control strategy was demonstrated experimentally on a physical biped.

CPGs are neural circuits which produce high dimensional locomotion patterns from low dimensional input signals. The use of CPGs in bipedal locomotion was inspired by Taga's seminal work [73, 74]. Taga et al. demonstrated 3D bipedal locomotion using CPG's for

robust and adaptive gait generation of a high-DOF robot [75]. An advantage of using CPG models is that it drastically reduces the dimensionality of the walking control problem and gait synthesis can be formed by generating a few higher level control signals [76]. A few key examples of bipedal locomotion with CPGs are [77, 78, 79].

2.2.5 Foot Placement

Recently, an alternative problem formulation focusing on restoring balance has been proposed. The Foot Placement Estimator (FPE), introduced by Wight et al. [8] formulates an approach to restore balance by controlling swing foot position during the gait cycle. Using the conservation of angular momentum, the FPE equation determines the location on the ground where the total energy of an unstable biped after swing foot impact is equal to the peak potential energy. If a step is taken before the FPE location, the post impact energy of the system causes the biped to fall over. Conversely, stepping beyond the FPE location on the ground causes the biped to fall back onto the hind leg.

The solution to the FPE equation itself can be used as a recovery mechanism (i.e. in the face of a destabilizing disturbance) with existing ZMP-based strategies. As shown in [8, 7], FPE can also be extended to form complete gait cycles to achieve dynamically stable walking. However, there are several key assumptions which are violated when attempting to implement this approach on a physical 3D robot. Namely, the theory assumes that the legs are massless and it only deals with the 2D dynamics in the sagittal plane.

The capture point (CP) concept, developed by Pratt et al. [80], is conceptually similar to the FPE. While the derivation of FPE is based on a simple compass biped model with fixed parameters, the CP theory was derived using complex motion models which included using a flywheel body to control/offset any disturbances through the use of rotational inertia. Ultimately, the simplicity of the model allowed the FPE theory to be extended to complete gait cycles, while the work presented by Pratt et al. simply solved the problem of lateral stabilization [8].

Englsberger et al. built on top of Pratt's CP concept [81] by developing two robust controllers based on LIPM dynamics. The CP end-of-step controller (CPS) responds to perturbations in real-time and adjusts the ZMP of the biped to shift the CP and regain stability. A second CP tracking (CPT) controller was also developed to realign the CP to its ideal trajectory if the biped experiences perturbations while walking. Both controllers were demonstrated in simulation and on the physical DLR biped. A similar approach was also developed by [82] which integrates the CP concept with Model Predictive Control (MPC). The MPC control scheme was developed to improve ZMP preview control for robustness

to strong perturbations [83]. The CP-MPC control scheme was also demonstrated on the DLR biped.

Recently, a more comprehensive approach using CP for foot placement and gait synthesis has been proposed. De Boer et al. [84, 85] focused on the ground/foot interaction to develop a robust and energetically efficient walking control strategy known as the capturability framework. Pratt demonstrated this strategy on the force-controlled compliant lower-body biped M2V2 [86, 87]. While the capturability framework is philosophically similar to the idea behind FPE, there are several key differences. The FPE approach uses simple local controllers to form complete gait cycles and can be used on position-controlled joints without any complex actuation systems. The capturability framework demonstrated in [85, 87] used separate controllers for the swing and stance legs whereas this approach uses a single global differential kinematic resolution for whole body motion control.

2.3 2D Foot Placement Estimator and Gait Generation

As the FPE forms the basis for subsequent development in this thesis, the 2D FPE theory [8, 7] introduced in Section 2.2.5 is reviewed in more detail in this section. Consider the standard compass biped model, shown in Figure 2.2. The physical parameters are the mass m, inertia about the center-of-mass I_{COM}, leg lengths L and leg separation angle β. Note that the angle θ_A is measured from the axis normal to the ground.

Now consider this system at the moment when the swing leg comes into contact with the ground at point B (the resulting behaviour is illustrated in Figure 2.3). The following assumptions are made in reference to the behaviour of the system around the pre-impact and post-impact stages to simplify the model for further analysis [7]:

Assumption 1 *There is an instantaneous transfer of balance (i.e. the stance foot at point A lifts up when the swing foot hits the ground at point B).*

Assumption 2 *The impact when the swing leg hits the ground at point B is assumed to be plastic (i.e. momentum is conserved but not kinetic energy).*

Assumption 3 *There is sufficient friction to prevent any slipping at the contact points.*

Assumption 4 *Gravity is assumed to be a non-impulsive force.*

Assumption 5 *The leg separation angle β is fixed in the double support phase.*

Assumption 6 *The legs are massless and therefore do not significantly alter the dynamics of the system.*

Note that assumption 5 implies that if both feet were to remain on the ground (i.e. double support phase), then by geometric symmetry about the normal, $\theta_A = \beta/2$.

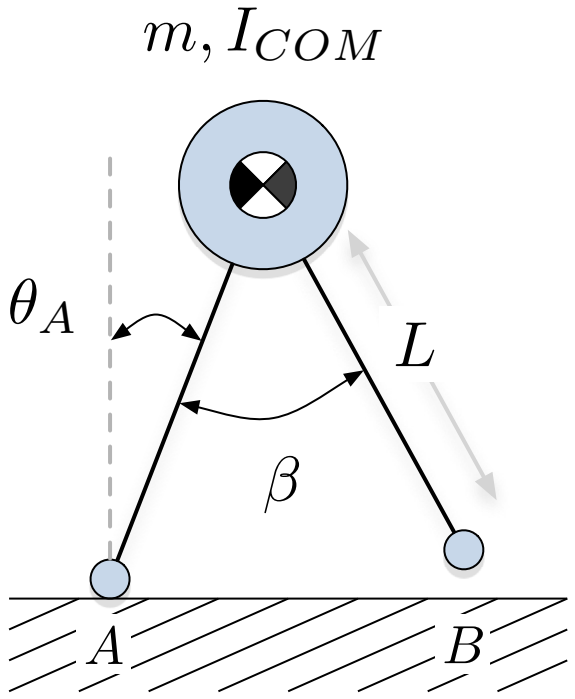


Figure 2.2: Standard compass biped model used for nonlinear analysis and derivation of the FPE equation.

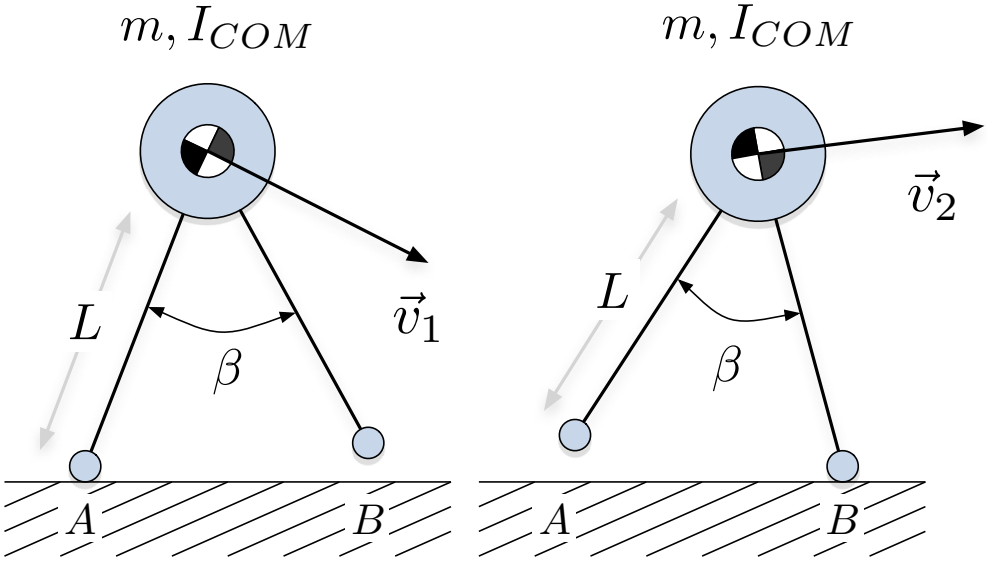


Figure 2.3: Compass biped model in the swing phase of the gait cycle in the pre-impact (left) and post-impact (post) configurations.

2.3.1 Equations of Motion

To formulate a state space representation of the biped, the equations of motion are derived as follows[1]. If the biped is treated as an inverted pendulum rotating about pivot point A (i.e. configuration shown in Figure 2.2), the equation of motion is found by applying Newton's second law:

$$\sum \tau_A = I_A \ddot{\theta}_A \tag{2.1}$$

$$\ddot{\theta}_A = \frac{mgL\sin(\theta_A)}{I_{COM} + mL^2} \tag{2.2}$$

This equation is only valid while the inverted pendulum swings about point A (i.e. pre-impact). When the swing leg comes in contact with the ground and based on the assumptions, point B becomes the new pivot for the inverted pendulum (post-impact). The motion of the post-impact system is now based on a different angular velocity (namely,

[1]Only key points of the derivation are summarized, details on the full derivation of the FPE equation can be found in [7, 88]

$\dot{\theta}_B$). However, the assumptions 1-4 also relate the angular momentum of the system in the pre-impact stage to the post-impact stage (Figure 2.3). Applying the conservation of angular momentum about point B, the post-impact angular velocity ($\dot{\theta}_B$) is a function of the pre-impact angular velocity ($\dot{\theta}_A$) [7]:

$$\dot{\theta}_B = \frac{(L^2 m \cos(\beta) + I_{COM})\dot{\theta}_A}{L^2 m + I_{COM}} \tag{2.3}$$

While pivoting about point B, if the biped does not have enough momentum to swing all the way through then it simply rocks back until swing leg A comes into contact with the ground. At this point, the equation of motion describing the system reverts back to (2.2). Given the geometric properties of the biped, it can be shown that the equation of motion about point B is given by:

$$\ddot{\theta}_B = \frac{mgL \sin(\theta_B)}{I_{COM} + mL^2} \tag{2.4}$$

Together, θ_A, θ_B, along with their derivatives completely describe the motion of the compass biped.

2.3.2 Unified State Equations

Assumption 5 imposes a geometric constraint which can be used to combine the variables which completely define the motion. Note that the leg separation angle β only has to be constant at the instant of impact. At this instance, the angles θ_A and θ_B can be expressed as:

$$\begin{aligned} \theta_A &= \theta + \frac{\beta}{2} \\ \theta_B &= \theta - \frac{\beta}{2} \end{aligned} \tag{2.5}$$

Where the angles θ_A, θ_B, θ and β are shown in Figure 2.4. The single (unified) variable θ can be formed by rearranging the equations of motion described in terms of θ_A and θ_B:

$$\ddot{\theta} = \begin{cases} \dfrac{mgL\sin(\theta+\beta/2)}{I_{COM}+mL^2} & \theta < 0 \quad (2.6a) \\ \dfrac{mgL\sin(\theta-\beta/2)}{I_{COM}+mL^2} & \theta > 0 \quad (2.6b) \\ \dfrac{mgL\sin(\theta+\beta/2)}{I_{COM}+mL^2} & \theta = 0,\ \dot{\theta} > 0 \quad (2.6c) \\ \dfrac{mgL\sin(\theta-\beta/2)}{I_{COM}+mL^2} & \theta = 0,\ \dot{\theta} < 0 \quad (2.6d) \\ 0 & \theta = 0,\ \dot{\theta} = 0 \quad (2.6e) \end{cases}$$

The system defined by (2.6) is used to investigate stability and subsequently derive the FPE equation. The following function represents the conditional equation of motion corresponding to a region of the state space (defined by θ and $\dot{\theta}$):

$$\ddot{\theta} := F(\theta, \dot{\theta}) \tag{2.7}$$

This unified variable is used to form a state space representation with the states as θ and $\dot{\theta}$:

$$\dot{\Theta} = \begin{bmatrix} \dot{\theta} \\ F(\theta, \dot{\theta}) \end{bmatrix} \tag{2.8}$$

2.3.3 Conditions for Stability

The notion of stability is explicitly defined by [7] as follows:

Definition 2.3.1 *The biped has fallen if $\dot{\theta} = 0$ and any other point other than the feet is in contact with the ground.*

Definition 2.3.2 *The biped is balanced if $\dot{\theta} = 0$ and it has not fallen.*

Definition 2.3.3 *The biped is stable if for a given set of initial conditions and no further energy input to the system, the biped eventually comes to a rest in an upright position. Once at rest, a sufficiently small, impulsive, nonzero external disturbance to the biped should result in motion that will eventually return to the same stable, balanced position.*

The only physical configuration that can be achieved by definition 2.3.2 is if the biped is in the double support phase. This implies that being balanced is mathematically equivalent to the system $\dot{\Theta}$ remaining at its stable equilibrium at the origin. The second part of definition 2.3.3 implies that stability in the physical sense is equivalent to the origin of system $\dot{\Theta}$ being asymptotically stable, since the biped should return to the same balanced position after small enough perturbations.

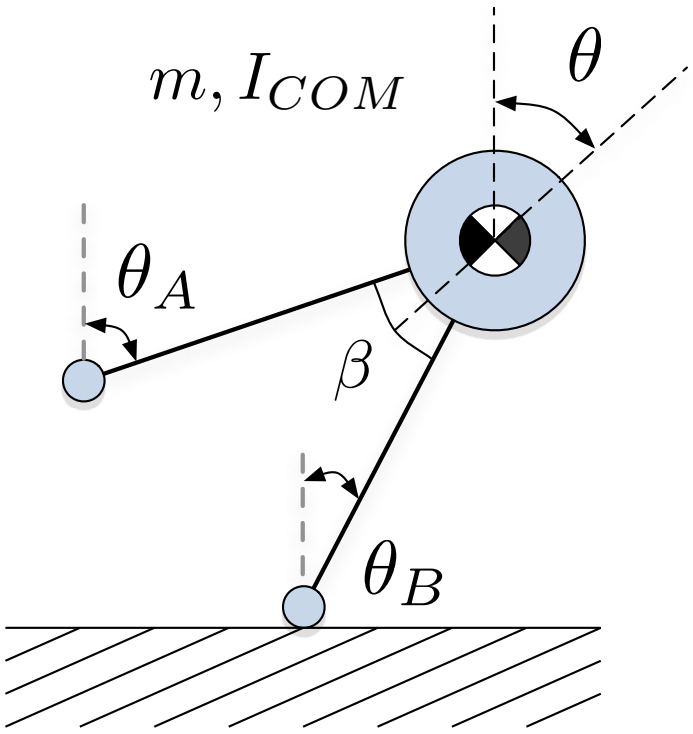


Figure 2.4: Unified variable θ used to simplify the analysis. It is easily observed that $\theta_A = \theta + \beta/2$ and $\theta_B = \theta - \beta/2$.

Thus, it is possible to determine the conditions for which the biped is stable if it can be shown that the origin of (2.8) is asymptotically stable. To this end, a Lyapunov function candidate based on the energy of the system ($U = T + V$) with an offset in potential energy is chosen to show asymptotic stability. For example, for the first equation of motion represented by (2.6) for $\theta < 0$:

$$V(\Theta) = \frac{1}{2}(I_{COM} + mL^2)\dot{\theta}^2 + mg(h - h_{datum}) \tag{2.9}$$

Where $h = \cos(\theta + \beta/2)$ and $h_{datum} = \cos(\beta/2)$. The Lyapunov candidate is positive definite if $-\beta/2 < \theta < \beta/2$ with the following $\dot{V}(\Theta)$:

$$\dot{V}(\Theta) = (I_{COM} + mL^2)\dot{\theta}^2 F(\theta, \dot{\theta}) - mgL\sin(\theta + \beta/2)\dot{\theta} \tag{2.10}$$

Analyzing the behaviour of $\dot{V}(\Theta)$ involves looking at each region of the state space specified by (2.6) and investigating the behaviour within the local region. In summary, $\dot{V}(\Theta) = 0$ for the cases where $\theta \neq 0$ (i.e. 2.6a and 2.6b) and is negative definite when $\theta = 0$, $\dot{\theta} \neq 0$ (i.e. 2.6c and 2.6d). Furthermore, the equilibrium point is the largest invariant set in:

$$E = \left\{\Theta | \dot{V}(\Theta) = 0\right\}$$

Thus, by the Barbashin-Krasovskii-LaSalle principle it is shown that the origin of $\dot{\Theta}$ is locally asymptotically stable in the sense of Lyapunov. In order to determine which initial conditions will exhibit a decaying orbit towards the origin, the exact boundaries of the local stability region are obtained by analyzing the behaviour of the total system energy with different initial conditions.

2.3.4 Computing the FPE Angle

Given that the system is asymptotically stable and the exact boundaries of the local stability are defined, it is possible to determine whether a specific location in the state space is stable in the sense of definition 2.3.3. However, the goal for FPE is to determine where the foot must be placed in order to restore stability, so the knowledge of the local stability is used to reformulate the problem.

To this end, the approach described in [7] introduces a parameter known as the FPE angle (ϕ). The projection from the COM at an angle ϕ to the ground surface provides

$$\frac{\left[mh(v_x\cos\phi + v_y\sin\phi)\cos\phi + I_{COM}\dot{\theta}_1\cos^2\phi\right]^2}{mh^2 + I_{COM}\cos^2\phi} + 2mgh\cos\phi(\cos\phi - 1) = 0 \tag{2.11}$$

the location of where the foot would need to be placed in order to restore stability to the unbalanced system. The actual solution to (2.11) yields the FPE angle ϕ, which can be obtained by using numerical techniques for solving non-linear equations. In essence, the FPE angle ϕ specifies the configuration to enter/remain inside the locally stable region of $\dot{\Theta}$ if it were to step in the *next* instant. By continuously tracking this angle while the biped is about to land the swing foot, the angle ϕ eventually converges to $\beta/2$ prior to impact.

The simple compass biped model made several assumptions which are now lifted. The leg separation angle β is no longer a fixed value throughout the whole gait cycle. Instead,

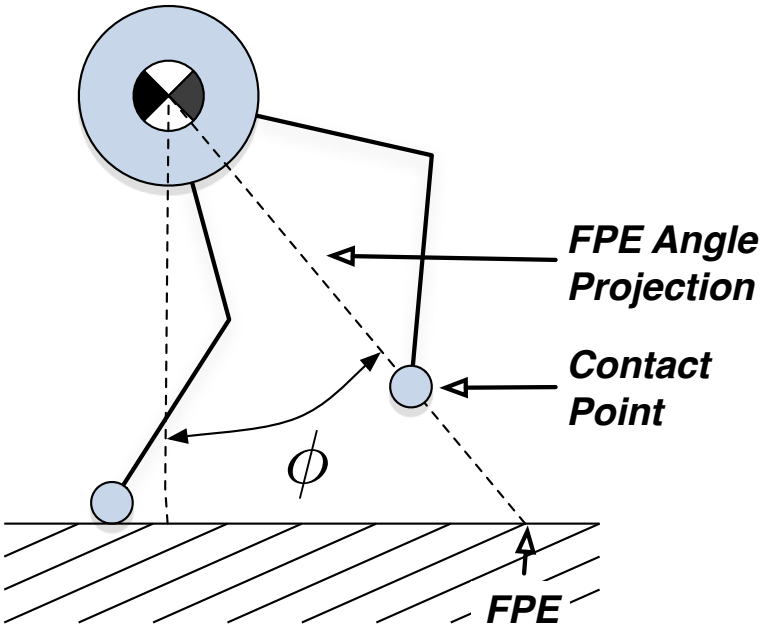


Figure 2.5: Graphical representation of the solution to the FPE equation for arbitrary robot configurations.

by assumption 6 that the legs are massless (and therefore they do not alter the dynamics of the system), the value of β can vary while the swing leg is being brought over to the contact point for the subsequent step. The solution to the FPE equation requires only that the leg length L and angle β are fixed at the instant of impact. Simply put, the compass biped model used in the derivation thus far only represents how a real biped looks when an impact occurs. During the swing phase, an arbitrary and more realistic model of the biped looks like that shown in Figure 2.5, where the angle ϕ, projection from the COM and the FPE location on the ground plane are visualized.

2.3.5 Stability Analysis

In order to show that stepping on the FPE point can restore stability to an unbalanced system, the unified state space model was implemented in simulation along with a numerical solver for the nonlinear FPE equation (2.11). Three experiments were devised to analyze the behaviour of the system when the swing foot steps exactly on, behind and ahead of the FPE location obtained from the solver. These three cases are labelled perfect, under and over stepping, respectively. The simulations assume arbitrary physical parameters for m, I_{COM}, etc. The compass biped model with the single state variable is simulated to illustrate the effects of overstepping and under stepping. The leg separation angle β is

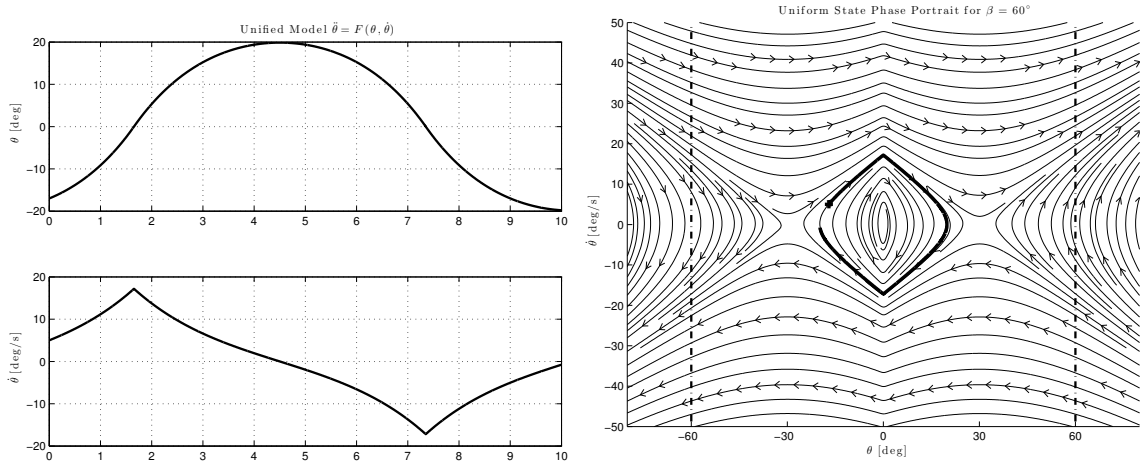


Figure 2.6: Time evolution and state trajectories for perfect stepping (i.e. foot lands extactly on the FPE point)

held constant and no energy is lost upon impact.

The results of the perfect stepping case shown in Figure 2.6 validate the efficacy of the FPE point on the ground. Starting from an unstable configuration, the simple biped model exhibits stability (i.e. rocking back and forth) due to no energy loss in the system by assumption. For a real physical biped, the energy losses experienced during impact would produce a decaying orbit towards the origin due to the asymptotic stability of the system within this local region.

In the under stepping case, the swing foot lands short of the FPE location on the ground. This behaviour results in an excessive energy and momentum which causes the biped to eventually fall over. Starting from the same initial conditions presented in the perfect stepping case, the results of understepping are shown in Figure 2.7. The time evolution and state trajectories exhibit unstable behaviour, validating the results presented in [8].

Lastly, the results of overstepping with the same initial conditions are presented in Figure 2.8. As expected, if the swing foot lands ahead of the FPE location, the system enters a stable orbit where the biped rocks back and forth. As mentioned previously, the energy losses experienced during impact for a real biped would cause this closed orbit to decay and eventually reach the equilibrium due to local asymptotic stability.

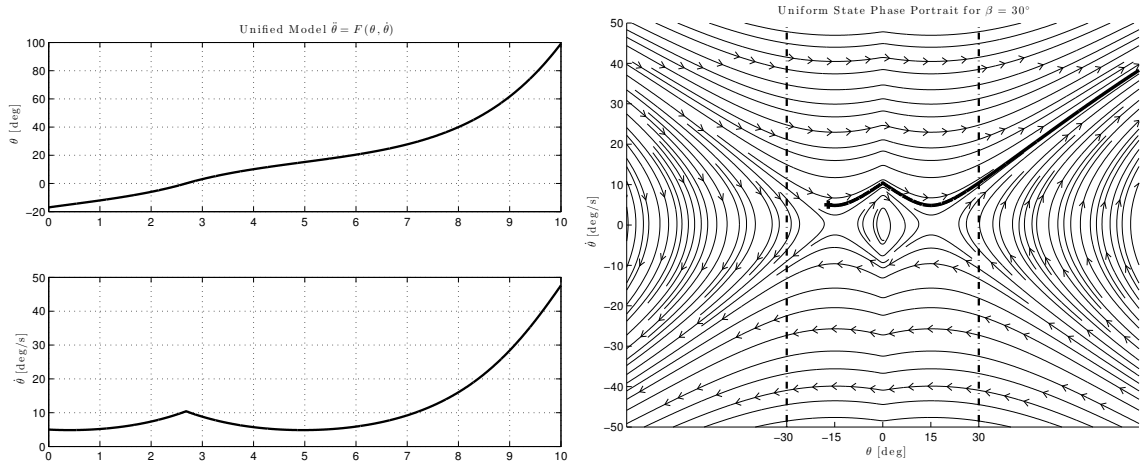


Figure 2.7: Time evolution and state trajectories for under stepping (i.e. foot lands behind FPE point)

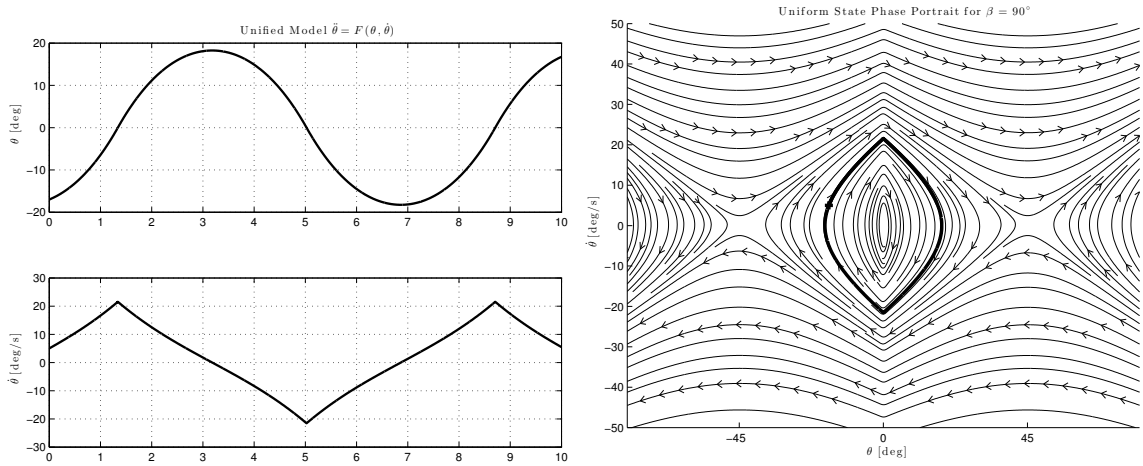


Figure 2.8: Time evolution and state trajectories for over stepping (i.e. foot lands in front of FPE point)

2.3.6 Forming Complete Gait Cycles

Wight used the FPE concept to develop full gait cycles using simple linear control techniques and a state machine [7]. Gait is initiated by destabilizing the robot in the desired direction of movement (forward or backward). Once destabilized, the FPE equation is solved numerically to obtain the FPE angle ϕ, which is used to provide the desired trajectory for the swing foot.

If continued forward progress is desired, the foot is commanded to precede the FPE. If no further forward progress is needed, the foot is commanded to the FPE. The desired trajectory is resolved to joint angles using inverse kinematics and implemented via joint level PD controllers. The complete state machine is shown in Figure 2.9. Due to symmetry, the states in Figure 2.9 can be reduced to **STAND**, **PUSH**, **LIFT**, **SWING** and **DROP**. For the remainder of this thesis, the sequence of state transitions from **PUSH** to **DROP** is referred to as the step cycle.

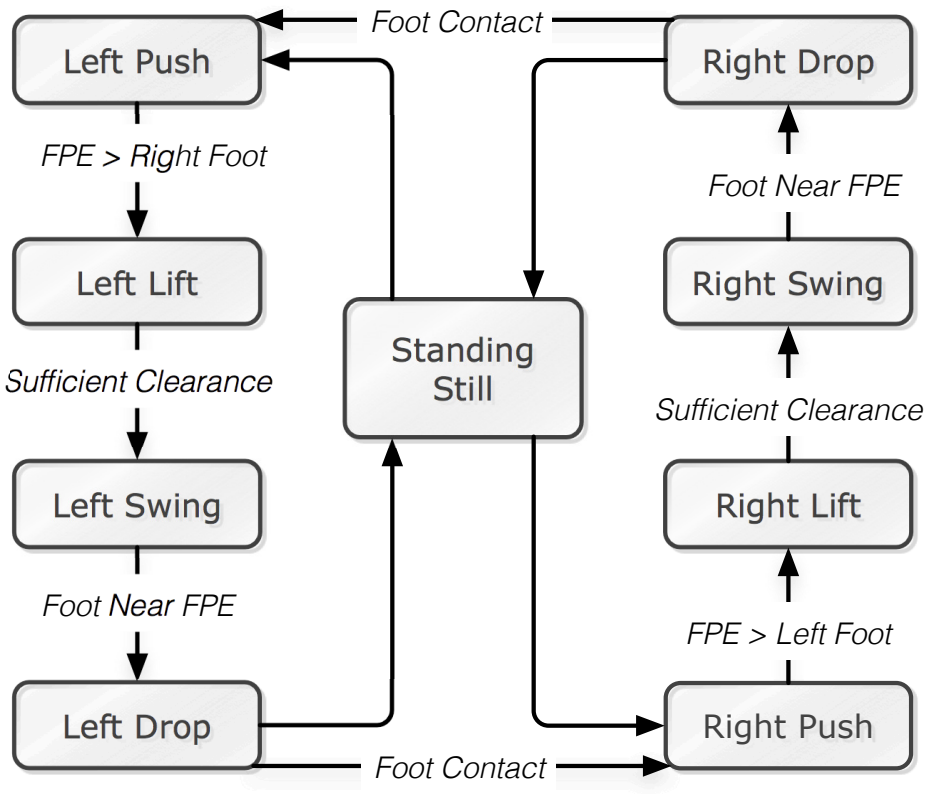


Figure 2.9: A simple state machine used in conjunction with the FPE algorithm to form complete gait cycles

.

Chapter 3

Electromechanical Design[1]

This chapter presents the electromechanical design of a bipedal robot for experimental validation. The goal was to design a lower body humanoid robot that would be easy to model in a dynamic simulator, manufacturable and capable of performing a wide variety of gait cycles. A fully actuated, kinematically redundant leg design was desired to enable walking, balancing and full mobility in the ankle (breakdown provided in Table 3.1). The bipedal robot is also designed to be fully actuated with overpowered electric motors to physically support the wide range of gait styles during experimentation.

As shown in Table 3.1, each leg is selected to have 7 DOF which provides a total of 14 actuated DOF for the bipedal robot. A serial link mechanism approach is used for its simplicity in design and kinematics. The bipedal robot is a fully powered active walker (i.e. every joint is actively controlled) with an electric motor drivetrain. DC motors with geared drives are selected because it is easy to model the actuator dynamics and achieve accurate joint level position control.

Table 3.1: Degrees-of-Freedom in each leg of the bipedal robot.

Joint	**DOF**
Hip	3
Knee	1
Ankle	3

[1]Part of the electromechanical design work in this chapter was completed as part of the ECE 499 project course at the University of Waterloo.

There are several other key design considerations which impact the performance of the finished product. The complete design process consisted of selecting the appropriate drivetrain components and designing the mechanical chassis. The primary design objective was to produce enough mechanical power for the robot to achieve walking. The secondary design objective was to keep the overall weight low and raise center-of-mass (COM) position as high as possible.

The initial design specification was obtained by modeling the system dynamics during an estimated gait cycle (Section 3.1). The selection of electromechanical components comprising the drivetrain system was used to address the primary design objective (Section 3.2). These drivetrain components include DC motors and appropriate gearing to provide the reduction ratio required to meet the initial design specifications. After the primary objective was addressed and the drivetrain components were selected, the secondary design objectives were satisfied by adjusting the mechanical structure and manipulating placement of components (Section 3.3).

3.1 Dynamic Modeling

A dynamic model of the proposed system must be developed and analyzed under normal operating conditions to determine the forces acting on the system and torques experienced at each of the joints. The forward or direct dynamic model provides a transformation from the joint torques ($\boldsymbol{\tau}$) to the resulting joint positions ($\mathbf{q}$), velocities ($\dot{\mathbf{q}}$) and accelerations ($\ddot{\mathbf{q}}$).

$$\ddot{\mathbf{q}} = f(\boldsymbol{\tau}, \mathbf{q}, \dot{\mathbf{q}}, \mathbf{F_C}) \tag{3.1}$$

$\mathbf{F_C}$ represents the contact force(s) felt at the feet. Given this model, a dynamic simulation environment can be designed to estimate the torque loading at different joints on the bipedal robot during the gait cycle. This estimate can form the basis of the initial design specifications from which motors and gearing can be selected. An important consideration in this model is the inclusion of contact forces during the stance phase of the gait cycle. The ground reaction forces exerted on the robot's feet have a significant impact on the dynamic requirements of each joint.

3.1.1 Equations of Motion

The general form of the equation of motion describing a n-DOF humanoid robot (with a floating base) is given by:

$$\mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \mathbf{J}^T\mathbf{F_C} \tag{3.2}$$

Where $\mathbf{A}(\mathbf{q})$ is the $(n+6) \times (n+6)$ inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ are the centripetal/Coriolis terms, $\mathbf{g}(\mathbf{q})$ is the gravity vector and $\mathbf{J}$ is the Jacobian matrix which provides the mapping between the joint and work space variables. The $(n+6) \times 1$ generalized force vector $\boldsymbol{\tau}$ is segmented as follows:

$$\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\tau}_{\mathbf{act}} & \boldsymbol{\tau}_{\mathbf{base}} \end{bmatrix}^T \tag{3.3}$$

Where $\boldsymbol{\tau}_{\mathbf{act}}$ represents the n actuated DOF and $\boldsymbol{\tau}_{\mathbf{base}} = \mathbf{0_{6\times 1}}$ are the remaining non-actuated DOF at the floating base. In order to incorporate the effects of contact forces at both feet of the biped robot, (3.2) is expanded as follows:

$$\begin{bmatrix} \mathbf{A_{11}} & \mathbf{A_{12}} \\ \mathbf{A_{21}} & \mathbf{A_{22}} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_{\mathbf{act}} \\ \ddot{\mathbf{q}}_{\mathbf{base}} \end{bmatrix} + \begin{bmatrix} \mathbf{b_1} \\ \mathbf{b_2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_{\mathbf{act}} \\ \mathbf{0} \end{bmatrix} + \mathbf{J_L}^T\mathbf{F_L} + \mathbf{J_R}^T\mathbf{F_R} \tag{3.4}$$

Where the generalized acceleration vector $\ddot{\mathbf{q}}_{\mathbf{act}}$ represents the n-DOF joint motion and $\ddot{\mathbf{q}}_{\mathbf{base}}$ represents the 6 DOF base motion. The gravitational and Coriolis terms are combined into the $\mathbf{b}$ vector. The $\mathbf{J_L}$ and $\mathbf{J_R}$ are 6 x n Jacobian matrices for the left and right legs, respectively. The force vectors $\mathbf{F_L}$ and $\mathbf{F_R}$ represent the contact force felt at the left and right foot, respectively. The method of calculating the magnitude of the force itself depends on the contact model used for the system. Equation (3.4) is used to represent the dynamics of the 14 DOF bipedal robot with $n = 14 + 6 = 20$ generalized coordinates.

3.1.2 Recursive Newton-Euler

The Recursive Newton-Euler (RNE) algorithm computes the inverse dynamics model recursively in order to arrive at a final solution for the direct dynamics [89]. The algorithm serially traverses through the kinematic chain to perform calculations using Newtonian mechanics. The first pass of the algorithm starts at the base link and traverses down the serial chains of the left and right legs calculating the kinematics. The second pass starts at the end of the serial chain(s) and works back up to the base link calculating the force

transmissions. After the final pass, the RNE algorithm produces the torques at each joint and the 6 dimensional forces and moments felt by the (floating) base joint. The RNE algorithm reformulates the equation representing the complete dynamics of the system (3.4) to obtain the direct dynamics model.

Kajita's toolbox [90] was used as a starting point for the direct dynamics model. The toolbox provides preprogrammed routines for common robotics algorithms (e.g. forward/inverse kinematics, etc). The kinematic and dynamic parameters of the robot are defined in a global structure variable called `uLINK`. The packaged scripts and functions access key

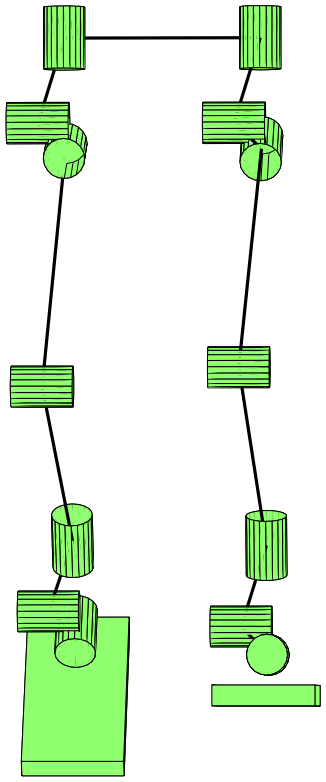


Figure 3.1: Screenshot of robot structure drawn from the uLINK definition generated directly from CAD.

information (e.g. kinematic hierarchy of links) from the global structure to perform the calculations. An external application was developed to interface with the CAD software package's application programming interface (API) to extract the kinematic and dynamic parameters directly. The application used the extracted data to automatically generate the CAD-equivalent `uLINK` representation in Matlab code (shown in Figure 3.1).

With the `uLINK` structure generated, any of the preprogrammed routines in the toolbox can be used. The toolbox provides a function `ForwardDynamics` to compute the direct dynamics model using the RNE algorithm. However, the calculations omit the effects of any external forces acting on the system. Since it is important to incorporate the contribution of the ground reaction forces, the toolbox was modified to model the complete dynamics of the system.

3.1.3 Contact Modeling

One challenging aspect of dynamic simulations is to accurately model the effects of external forces on the system during the stance phase. To emulate the ground reaction forces during

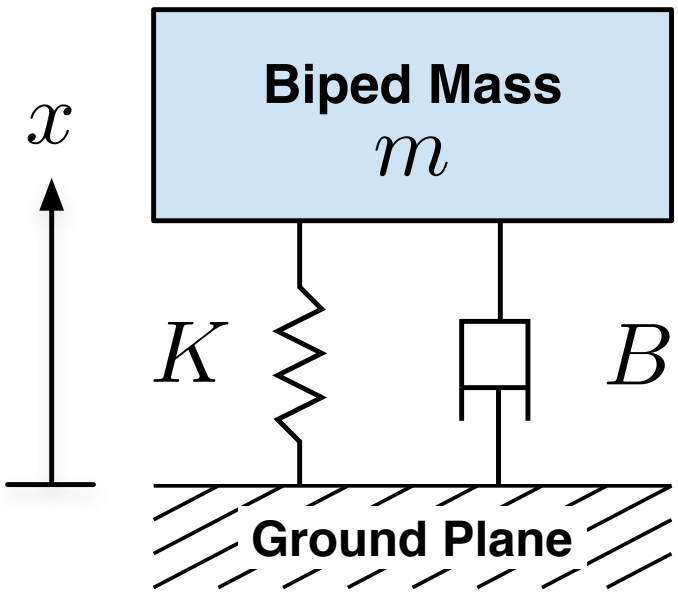


Figure 3.2: System diagram of spring-damper contact model used in dynamic simulations.

this period of the gait cycle, a contact model is used to generate an approximate contact force. For the initial design specifications, a spring-damper contact model is used (shown in Figure 3.2). Both feet are modelled as point contacts which experience a normal force proportional to the output of the spring-damper system when in contact with the ground. A more complex version of (3.5) is used for full dynamic simulations in later chapters (Section 5.2.1).

To incorporate the effects of ground contact into the existing dynamic model, the resulting force produced by (3.5) is injected into the RNE algorithm between the end of the first pass and the start of the second pass. At the start of the second pass, the external contact force is used to initialize the backwards recursion.

$$F_{normal} = B\dot{x} + Kx \tag{3.5}$$

The contact force proportional to the output of the spring-damper system (3.5) was used to initialize the backwards pass as shown in Listing 3.1:

Listing 3.1: Contact force injection in RNE backwards recursion.

```
if uLINK(RCONTACT).p(3) < 0.0
        Kr = K; % Spring Constant
        Br = B; % Damper Constant

        % Velocity of the Contact Point in World Frame:
        vr = uLINK(RCONTACT).vo + cross(uLINK(RCONTACT).w, uLINK(RCONTACT).p);

        % Normal Force Exerted by Spring-Damper Contact Model:
        N = -Kr*uLINK(RCONTACT).p(3)-Br*vr(3);

        % Inject Normal Force Into RNE Backwards Pass Initialization:
        FCR = [0 0 N]';
else
        FCR = [0 0 0]';
end
```

The variables `Kr` and `Br` represent the spring and damper coefficients in the spring-damper contact model. `RCONTACT` is a constant link number representing the point contact foot. The velocity due to the point foot coming in contact with the ground (`vr`) is recomputed by considering the original velocity (`vo`) and the cross product between the position of the foot (`uLINK.p`) and angular velocity (`uLINK.w`). The resulting normal force is computed and

injected into the backwards pass if the foot position is below the ground level (condition shown in line 1). With this modification included, the complete direct dynamics can be used to obtain the joint torques during an estimated gait cycle for the initial design specification.

3.1.4 Gait Estimation

In order to emulate the operating conditions for the 14 DOF bipedal robot, the dynamic simulation must include all dynamic effects experienced during a complete gait cycle. One method of emulating the normal operating conditions is to use human gait as an estimate. While it is ambitious to expect a bipedal robot to match the performance of human-like walking, it provides a basis from which the initial selection of electromechanical components can be made. The goal is to extract the kinematic parameters experienced by the human limbs while walking and impose similar conditions on the robot in dynamic simulation. Given these constraints, the inverse dynamics problem can be solved to obtain the joint torque estimates during various stages of the gait cycle. Coupled with the modifications to emulate contact forces during the stance phase, this approach provides a fast method of estimating the torque loading at each joint to develop the initial design specification.

In order to use kinematic parameters from human gait, data published from the 2008 Dynamic Walking conference was used [91]. The published dataset used sensors to measure the joint angles and velocities of lower body human limbs under several gait cycle speeds (fast, slow and normal walk). This raw sensory data was imported into the Matlab environment using a 3D motion analysis toolbox BodyMech [92]. This information was incorporated into the dynamic simulation by enforcing the joint angles, velocities and accelerations from each captured frame of the reference dataset. The forced joint parameters were assigned to the global `uLINK` structure to be used in conjunction with the RNE-based inverse dynamics algorithm.

A simulation was implemented in the Matlab environment using a combination of the BodyMech toolbox and Kajita's toolbox. The BodyMech toolbox was used to extract the frame-by-frame 3D kinematic parameters as a human executed the gait cycle. These parameters were processed and stored in order to be played back on the robot model defined by the `uLINK` structure. To compensate for the differences in size between humans and the small biped, the joint velocities and accelerations were scaled by selecting a different reference gait speed from the published data set. Once the parameters were applied, the RNE-based inverse dynamics algorithm was used to determine the torques at each joint in the structure during the estimated gait. This process was repeated for each frame (illus-

Figure 3.3: Frame captures of the uLINK structure under the estimated gait cycle to obtain initial design specifications.

trated in Figure 3.3) in the captured dataset and the results of robot's kinematics/dynamics ($\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ and $\boldsymbol{\tau}$) were tabulated for post processing.

3.1.5 Initial Design Specifications

The results of the dynamic simulations under human-gait analysis form the basis for the initial electromechanical components selection process. As expected, it was observed that the largest demands in torque loading on each of the joints occurred during the stance phase immediately after the heel strike. The torque demands at that instant spike immediately to almost three times the steady state torque loading on the joints.

Selecting motor specifications based only on the peak torque requirements would result in large and overpowered actuators for the continuous operational loads. The secondary design objective is to keep the overall system mass low, making it desirable to minimize the actuator size since it contributes to the majority of the weight in the system.

Motor manufacturers specify rated conditions for *continuous* operation, while the peak torque demands in this application only exist for a short period of time when ground contact is made. It is possible to safely push DC motors beyond the nominal rated capacities for short periods of time by considering the maximum mechanical power output and thermal characteristics. Therefore, it is desirable to select motors primarily on the steady state

torque load requirements while investigating other characteristics to ensure that they can handle the peak load requirements.

Table 3.2: Individual joint torque demands for each DOF in one leg using gait estimation.

Joint	**Normal**	**Peak**
Hip Yaw	3.698 Nm	3.698 Nm
Hip Roll	13.606 Nm	14.054 Nm
Hip Pitch	10.081 Nm	38.295 Nm
Knee Pitch	5.229 Nm	16.539 Nm
Ankle Yaw	3.858 Nm	3.858 Nm
Ankle Pitch	4.395 Nm	7.923 Nm
Ankle Roll	2.824 Nm	5.021 Nm

While the torque demands vary for each joint (see Table 3.2), the results from the analysis are concatenated to obtain a single set of design specifications. This reduces the number of different models selected from the manufacturer and it also simplifies the electromechanical design process. For the initial design, the normal torque specification is obtained by taking an average of joint torque values during the swing phase. Since the ground reaction forces experienced during the stance phase produce large torques in a short period of time, the peak torque specification is obtained by selecting the highest torque value from all frames. The combined initial design specifications for the drivetrain components are summarized in Table 3.3.

Table 3.3: Initial design specifications for drivetrain components.

Condition	**Torque**	**Velocity**
Normal	14 Nm	2.48 rad/s
Peak	50 Nm	6.82 rad/s

The contact model parameters (spring and damping coefficients) were observed to have a noticeable effect on the peak design specifications. Since the contact model only provides a crude approximation of the ground reaction forces that the real robot will experience, the final selected value was chosen to provide a peak torque which averages between the largest and smallest values. The final hand-tuned spring and damping coefficients used in deriving the design specifications were $K = 10000$ and $B = 100$, respectively. These

contact model parameters result in a steady-state ground penetration depth of $0.25cm$. The foot characteristics can be modified to match this behaviour by adding a rubber sole.

3.2 Drivetrain Selection

The drivetrain selection process includes selecting the appropriate components to transmit the required torque (as per the initial design specifications) to each joint. The electric machines used to generate mechanical power for the purposes of this project are coreless DC motors. The motors used for the 14 DOF bipedal robot were supplied by Micromo Solutions (part of the Faulhaber Group) [93]. The online product catalogue was used as a reference tool for the drivetrain selection process. The majority of the weight contribution of the complete system is attributed to the motors. Therefore, the design objective was to select a combination of motors and appropriately sized gearheads to meet the design specifications while keeping the overall mass of the system as low as possible. The remainder of this section presents the selection of appropriate motor and gearhead combinations to meet the design objectives.

3.2.1 Mechanical Power Requirements

To narrow down the selection process into a subset of the motors which meet the initial design specifications, preliminary mechanical power calculations are performed. The mechanical power output required by each generator is specified by the required torque (τ) and speed (ω) of the motor:

$$P = \tau \times \omega \tag{3.6}$$

The initial specifications for the drivetrain components listed in Table 3.3 list two values for the torque. As mentioned previously, designing the system to meet the peak torque requirements would yield significantly larger motors and increase the overall mass. Instead, it is desirable to design the system to meet the average (continuous) torque requirements and ensure that the system is capable of withstanding the large impulsive forces experienced at the impact points of the gait cycle. Therefore, the initial mechanical power requirement calculation is based on the normal or average torque loading on the joints (14Nm). While no speed requirements were specified in the initial design specifications, the tabulated kinematic and dynamic data collected from the dynamic simulation was used

to compute the average speed experienced by the joints and the largest value was selected for the purposes of calculation. This provides an additional buffer in the mechanical power considerations since not all joints are likely to experience the highest speed. The average speed calculated from the dynamic simulation was found to be 65RPM. Substituting these values along with the conversion factor for units:

$$P = \tau \times \omega = 14Nm \times 65RPM \times 0.1047 = 95.28W \tag{3.7}$$

Note that due to the assumptions made in calculating this value, 95.28W represents an upper bound on the amount of power the design should be capable of providing. Therefore the potential solution candidates should be rated for the nominal power around the region of this value. The analysis tools presented in the following sections were used to evaluate the remaining actuation solution candidates.

First, DC motors were analyzed based on their torque-speed (Section 3.2.2), power and efficiency (Section 3.2.3) and thermal (Section 3.2.4) characteristics. Full DC motor plots are shown for the final selection since the selection of gear ratio was treated as a separate design decision. With the appropriate motor selected, compatible gearhead solutions were analyzed to select the reduction ratio which met the design specifications (Section 3.2.5).

3.2.2 Torque-Speed Analysis

The primary tool used to analyze and compare alternative solutions is the torque-speed graph. Analysis of this graph is used to address the peak requirements. Given the structure of the motor specifications provided by Micromo Solutions, a spreadsheet was constructed to automatically generate the torque-speed characteristics from the motor constants. These constants were pre-populated from the manufacturer specification sheets and the torque-speed graphs were automatically generated.

$$\omega = \omega_{no-load} - k\tau \tag{3.8}$$

The relationship between the torque of the motor and the speed of the revolving shaft is approximately linear so it is typically specified by a slope constant for the curve. While the constant k is usually provided as a positive value, the slope of the torque-speed curve is negative (i.e. as the torque increases the speed decreases). The full torque-speed profile of the final motor selection is shown in Figure 3.4.

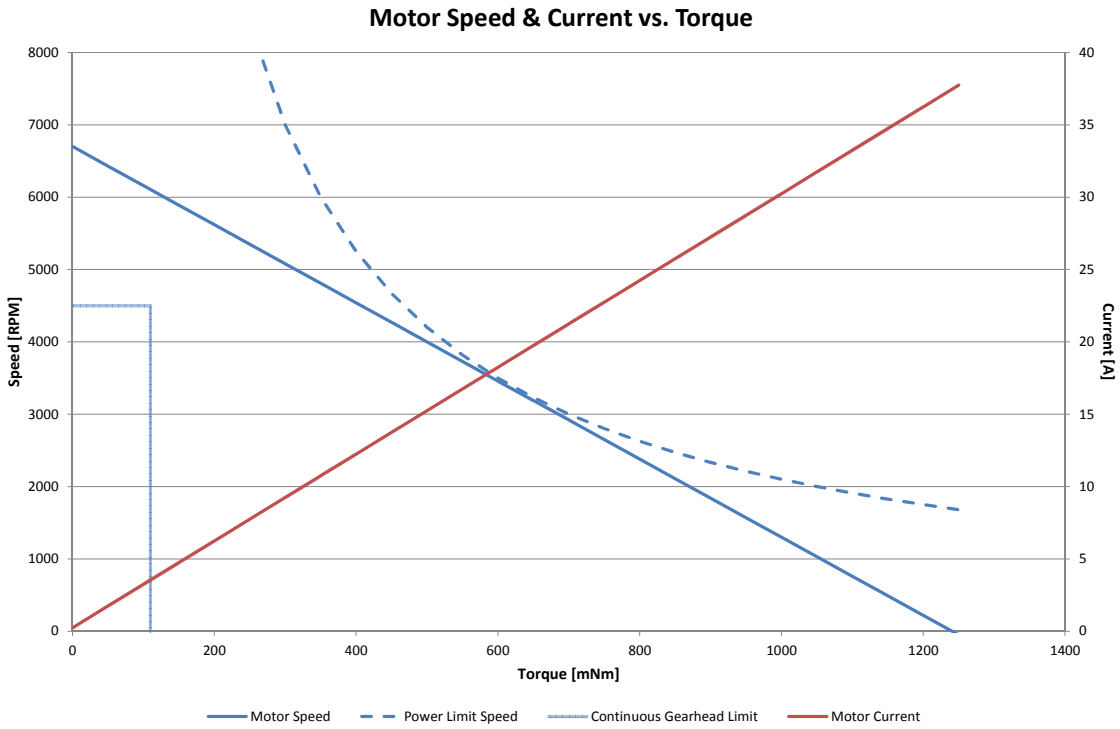


Figure 3.4: Micromo 3257CR Coreless DC Motor: Speed and Current vs Torque

3.2.3 Power and Efficiency Analysis

Another key factor in comparing drivetrain components is the relationship between the power produced and the overall efficiency of the motor as a function of its torque. The efficiency curve compares the ratio of electrical power input to mechanical power output. While it is typically desirable to operate the motor at its peak efficiency, the torque demands fluctuate throughout the gait cycle making it difficult to base the motor selection process on this basis alone. However, it is desirable to ensure that the motor is reasonably efficient within the expected range of operation. The power and efficiency profile of the final motor selection is shown in Figure 3.5.

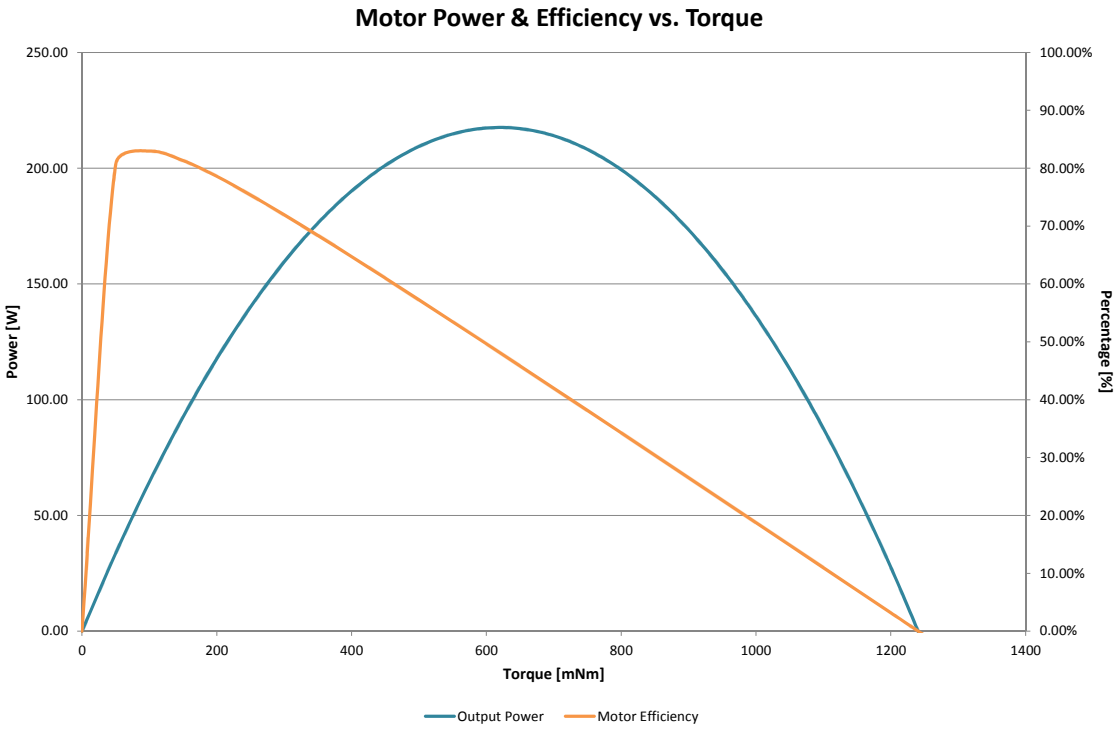


Figure 3.5: Micromo 3257CR Coreless DC Motor: Power and Efficiency vs Torque

3.2.4 Thermal Analysis

As mentioned previously, out of the initial design specifications the motor selection process is based on the continuous or average torque loading on the joints. However, during each gait cycle the torque demands peak due to external, impulsive force interaction with the environment (i.e. heel-strike). While these peaks may be outside of the motor's continuous operating range for a very short period of time, the heat generated inside the motor core can build up and cause significant damage. Another consideration is the effect of thermal build up of the motor under continuous use at the specification of 14Nm. Therefore it is necessary to generate thermal plots alongside the torque-speed analysis to verify that the selected motors can withstand the temperature rise during operation. Given an assumed value for the ambient temperature, the total temperature of the motor can be calculated as:

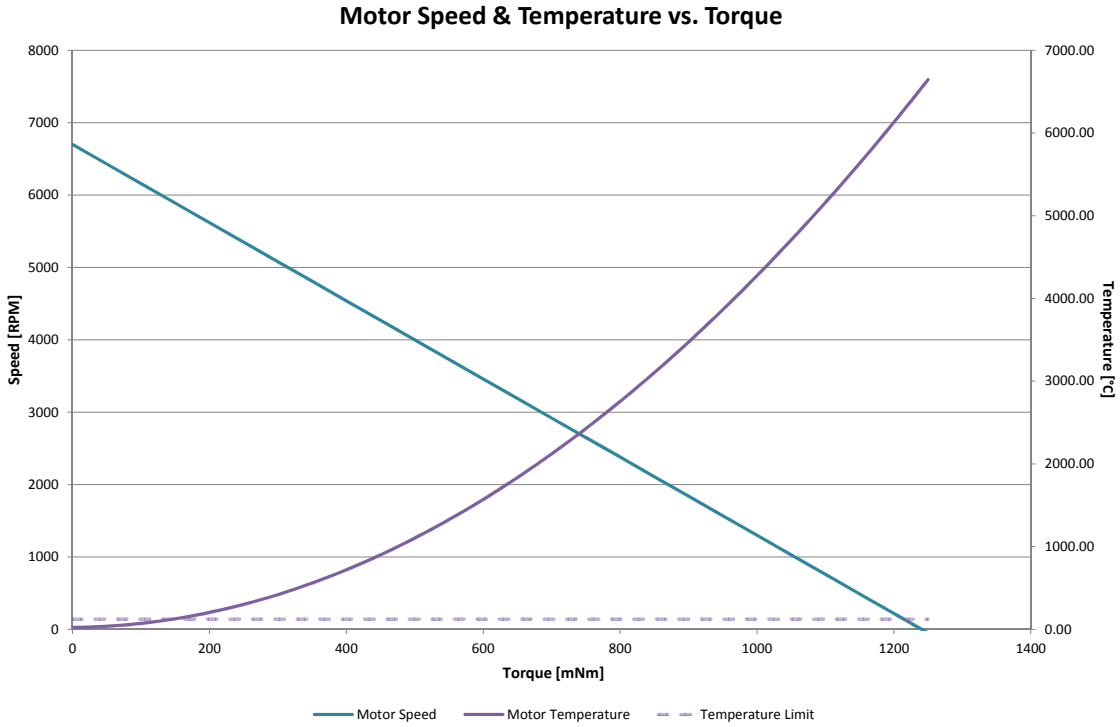


Figure 3.6: Micromo 3257CR Coreless DC Motor: Speed and Temperature vs Torque

$$T_{total} = T_{ambient} + T_{inc} \tag{3.9}$$

Where the temperature increase inside the motor core is provided by the following relationship[2]:

$$T_{inc} = I^2 R \times (R_{th1} + R_{th2}) \tag{3.10}$$

I and R are the current through and the resistance of the motor windings, respectively. Constants R_{th1} and R_{th2} are specified by the motor manufacturer. The thermal characteristics of the final motor selection are shown in Figure 3.6.

[2]The thermal characteristic equations were obtained from the manufacturer's technical documents on motor calculations available at `http://www.micromo.com/motor-calculations.aspx`

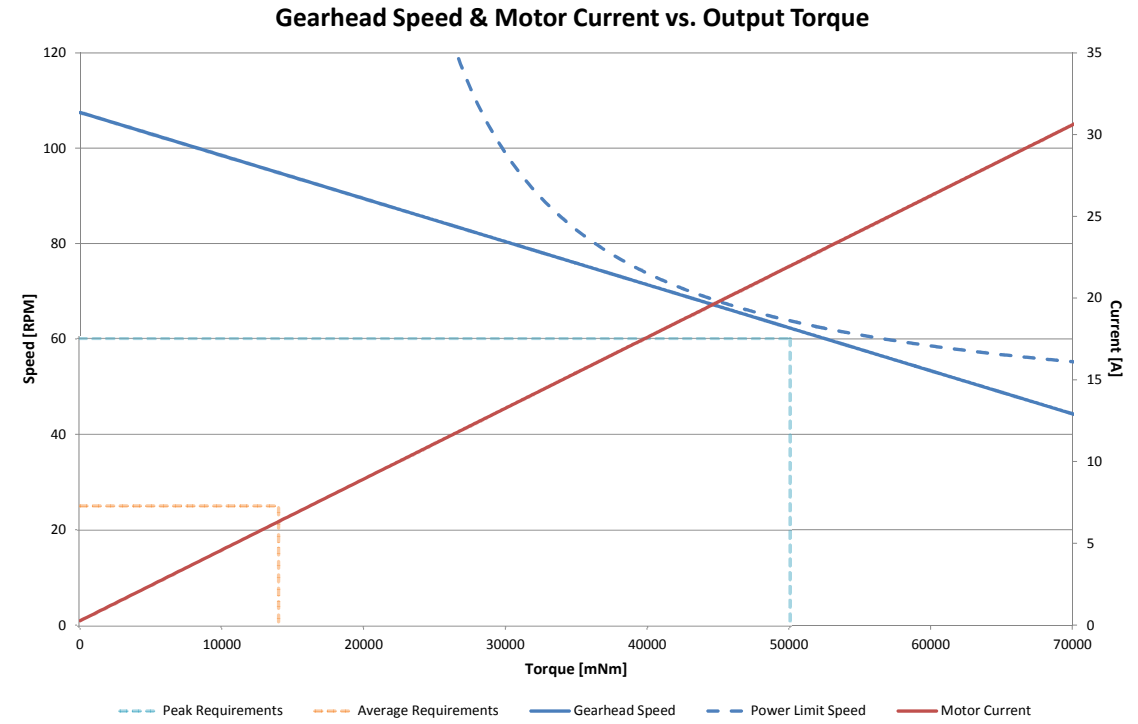


Figure 3.7: Micromo 38A Precision Gearhead: Speed and Current vs Torque

3.2.5 Final Configurations

Aside from the torque-speed, power, efficiency and thermal analyses, there are several other considerations which factor into the design while selecting drivetrain components. These considerations pertain to the secondary objective to minimize the overall weight of the system while simultaneously shifting the COM higher. Between the motor and the gearhead, the majority of the weight comes from the actual coreless DC motor so it is desirable to select a smaller (lighter) motor and meet the specified torque requirements by increasing the gear reduction ratio. Another consideration is the physical size of the motor itself, as the diameter and length of each motor vary significantly between different offerings. Since gearheads are coupled directly onto the motor shaft, the complete drivetrain assembly is usually long.

Given these considerations and the analysis in the previous sections, a coreless DC motor from the Micromo Solutions catalog was selected (Model #: 3257CR). This motor achieved

all of the primary design considerations (average torque requirements, operating within a stable thermal region). Coupled with the motor is a precision gearhead which is capable of producing torques slightly above the initial design specification requirements. The torque-speed characteristics at the gearhead shaft (as shown in Figure 3.7) are used to verify that the mechanical power output meets the design specification for normal operation.

While it is desirable to limit the number of different motor and gearhead combinations, it was determined that this combination of drivetrain components was vastly overpowered for the very last joint on the leg (ankle roll). A second combination of motor and gearhead was selected specifically for this joint to reduce the overall mass and raise the COM while meeting the lower joint requirements. This approach improves the controllability of the final system while limiting the total number of different design configurations. The final higher and lower power configurations are defined in Tables 3.4 and 3.5, respectively.

Table 3.4: Higher configuration of motor and gearhead combination from Micromo.

Model	**Mass**	**Length**	**Stall Torque**	**Max Power**
3257CR 12V Motor	242g	57mm	0.547Nm	84.50W
38A 240:1 Gearhead	410g	80mm	20.0Nm	-

Table 3.5: Lower configuration of motor and gearhead combination from Micromo.

Model	**Mass**	**Length**	**Stall Torque**	**Max Power**
3242CR 12V Motor	172g	42mm	0.193Nm	27.30W
32A 68:1 Gearhead	240g	57mm	6.0Nm	-

3.3 Mechanical Design

The mechanical design of the 14 DOF bipedal robot starts from the initial motor selection presented in the previous section. The 3D CAD models for each motor and gearhead combination are obtained directly from the manufacturer. The design goals for the mechanical chassis are to produce a solid light weight structure which is capable of withstanding external forces from the environment during gait and the applied torques from each joint. The secondary design objective of minimizing the overall weight while shifting the COM higher must also be addressed through the mechanical design.

3.3.1 Anthropometric Dimensioning

The goal for the 14 DOF bipedal robot is to achieve walking with human-like gait. To this end, anthropometric dimensioning is used as a starting point for the mechanical chassis design. Starting from a soft height requirement of the full lower body, the relative lengths are obtained for the main linkages (e.g. thigh, shank) from the anthropometric resource shown in Figure 3.8.

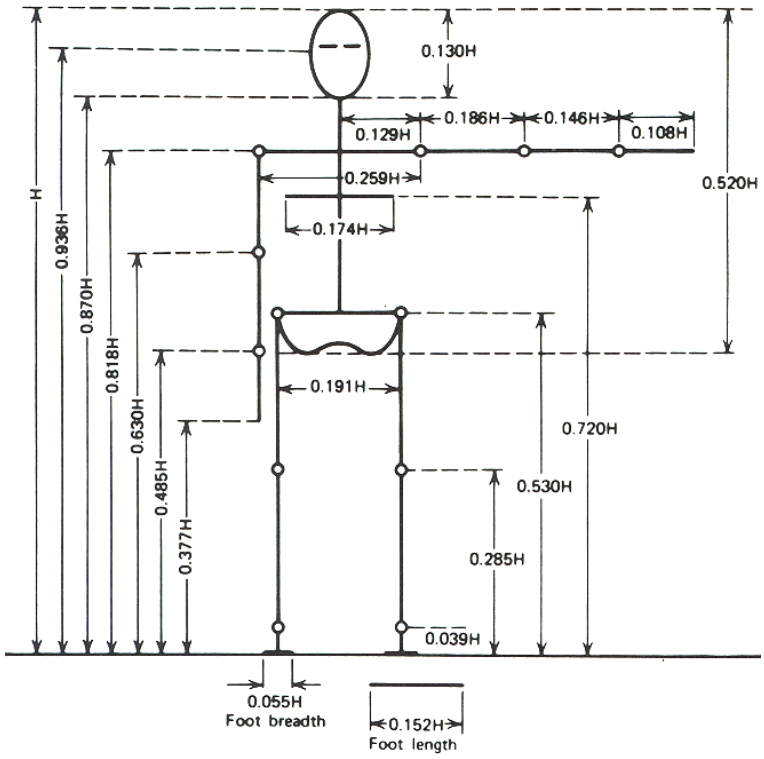


Figure 3.8: Resource for anthropometric dimensioning of lower body segments [94].

Given the ratios between major body segment lengths, a rough estimate for each of the chassis segment lengths is derived and summarized in Table 3.6. Note that these dimensions were only used as a starting point. The overall length of the drivetrain components (i.e. motor and gearhead combination) is an additional constraint which can limit how close the final chassis design is to the estimated segment lengths.

Table 3.6: Estimated segment lengths based on anthropometric dimensioning

Segment	**Height [mm]**
Hip	11.32
Thigh	277.34
Shank	278.47
Foot	44.15

3.3.2 Chassis Structural Design

The chassis is designed as a light weight but rigid frame to house the drivetrain components (actuators, gears, etc). The material selection and structural design of the chassis have an impact on the walking performance of the bipedal robot. For example, a light weight and structurally sound material like carbon fiber can be used to reduce the overall leg mass and improve the performance of the system during the swing phase. However, the major drawbacks to using advanced polymers like carbon fiber are the cost and manufacturability. As with most engineering design processes, a trade-off decision must be made in order to arrive at a solution which balances performance and feasibility.

Aluminum was selected as the light weight rigid material to compose the mechanical chassis for the 14 DOF bipedal robot. The decision was made based on the widespread availability of aluminum stock in various form factors (e.g. channels, beams, etc) and its relatively light weight when compared to other metal alloys. More specifically, Aluminum 5052 was selected due to its high fatigue strength. This alloy is also easy to work with for manual machining or computer numerical control (CNC) machine tools.

The low density of aluminum also helps address the secondary design objective of reducing the overall weight. However, the primary weight contribution to the overall system comes from the drivetrain components. This limits the flexibility for weight manipulation from a purely chassis design standpoint. However, mechanical design techniques can be used manipulate the COM location by strategically positioning the mounting points for motors and gearheads.

The amount of stock material used in the frame itself also impacts the overall weight and structural rigidity. For example, a link segment designed from a C-Channel beam is structurally sound (since it is a single solid piece of metal). However, it may be significantly heavier than a link segment composed of multiple pieces. The trade-off between minimizing the overall weight and structural rigidity is particularly important for the swing phase, when the additional structural rigidity reduces the vibrations in the lower body.

The initial chassis design of the legs for the 14 DOF bipedal robot was based on using C-Channel beams in the thigh and shank linkages. However, aluminum stock tolerances made it difficult to manufacture the initial chassis design with high accuracy. The C-Channel design was replaced with a ladder-like structure with parallel plates and cross braces to improve the manufacturability. The revised design was easily manufacturable with CNC machine tools and reduced the overall weight of the legs (compared to the initial C-Channel design).

3.3.3 Mechanical Power Transmission

The biped consists entirely of rotating mechanical joints between link segments. A common approach to shifting the COM position higher is to relocate motors higher in the lower body

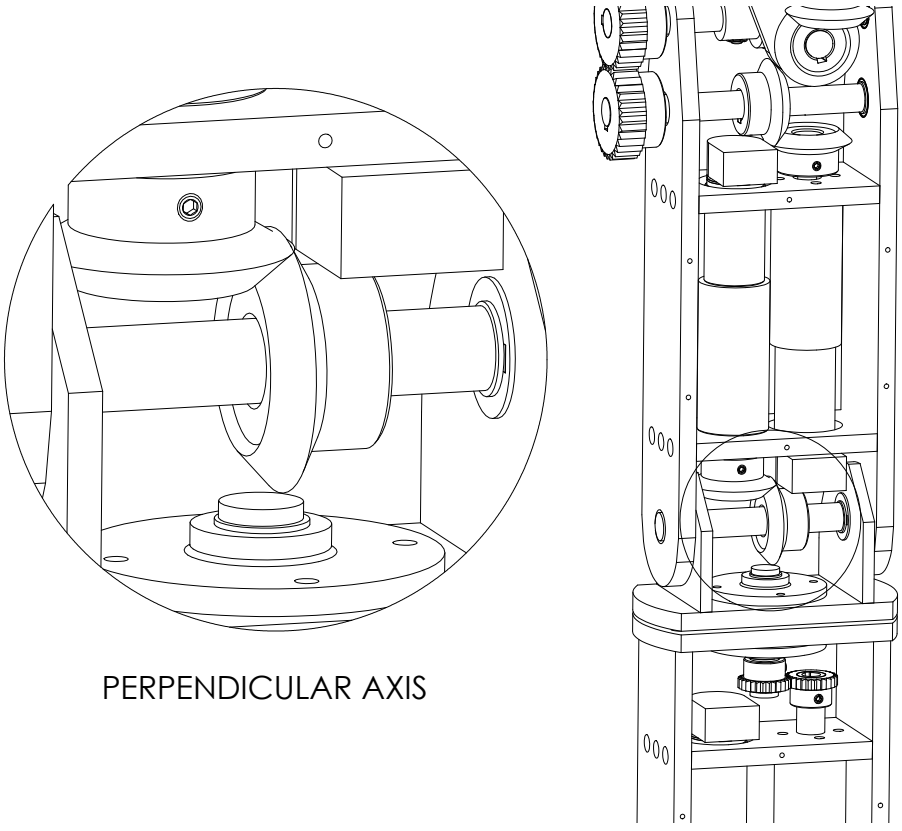


Figure 3.9: Perpendicular mechanical coupling to shift weight distribution.

structure and couple the actuator output to drive the joints through mechanical power transmission components. Some existing electromechanical designs for the lower body use belts or pulleys to transmit the mechanical power from the actuator output to the joint axis [24, 95, 96, 97]. However, as discussed in Section 2.1, the drive flexibility introduced by these systems adds additional complexity which is difficult to model and often requires tuning.

Instead, rigid machine components such as shafts, bearings and gears are used to transmit the rotational mechanical power between links in the 14 DOF bipedal robot design. This approach simplifies the electromechanical design and produces a system which is easy to model in simulation. By using components such as miter gears, the mechanical power transmission is shifted with a perpendicular offset (90°). This technique is used to strategically decouple the mounting position of a drivetrain assembly (i.e. motor with fixed gearhead) from the axis of rotation and relocate motors such that the weight distribution is manipulated in favour of having a higher COM position (illustrated in Figure 3.9).

Mechanical coupling and power transmission are also used to design intersecting axes of rotation. By using spur gears with a 1:1 reduction ratio, the output shaft of a drivetrain assembly can be positioned in a parallel offset (illustrated in Figure 3.10).

Shafts are frequently used to enable rotation between two link segments in the mechanical design. Ball bearings are common machine components which support the rotational movement of these shafts within a solid frame chassis. Traditional shielded ball bearings were selected for joints to support radial loads while permitting the rotation of a shaft on the inner ring (as illustrated in Figure 3.11). The double shielded construction provides protection against debris from entering the bearing assembly without producing significant internal friction forces that are present in the case of sealed bearings.

For yaw joints which actuate along the vertical axis, the impact forces experienced during the gait cycle exert an axial load which is typically not supported by traditional ball bearings (would cause the inner ring to "pop out"). Furthermore, these joints experience axial loads in one direction while the robot is standing on the ground but the load reverses direction if the robot is picked up by the torso. Therefore, bidirectional support for axial loads is required for the hip and ankle yaw joints. For these sections of the lower body, a combination of thrust and ball bearings are used to support radial and axial loads (as shown in Figure 3.12).

The complete mechanical CAD drawings of the 14-DOF bipedal robot can be found in Appendix A.1.

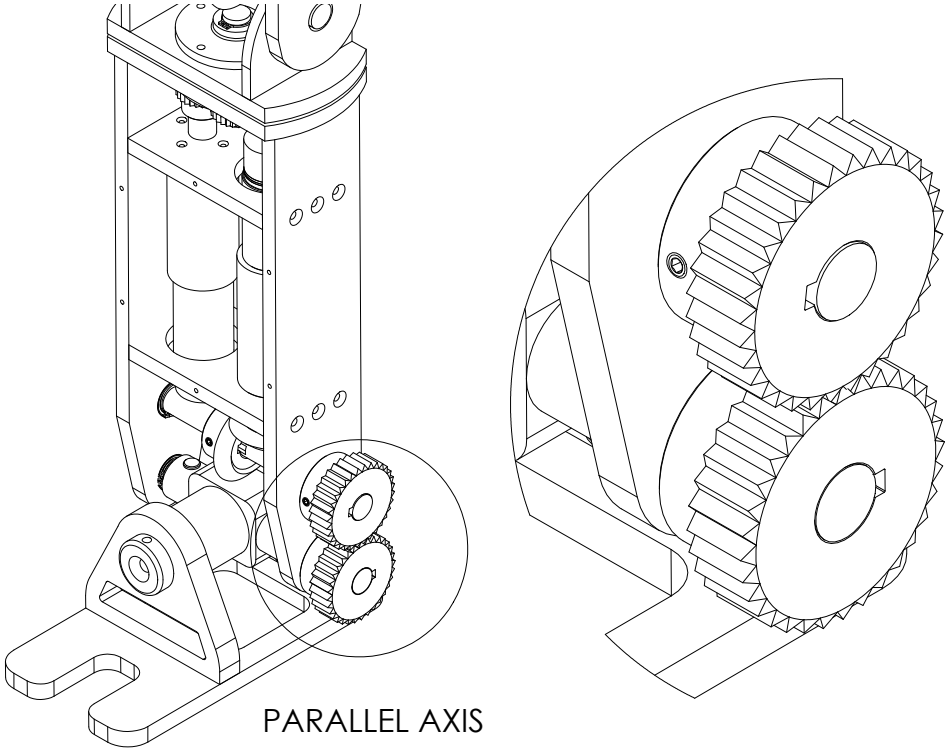


Figure 3.10: Parallel mechanical coupling to allow intersection axis of rotation.

3.4 Summary

The electromechanical design of the 14 DOF bipedal robot had two primary design objectives. The first was to produce enough mechanical power for the biped to achieve walking. The secondary design objective to keep the overall weight low and raise the COM improves the controllability of the final design. The initial design process started with basic dynamic modeling of the lower body. The forward and inverse dynamic models computed with the RNE algorithm provided a mathematical framework for estimating the torque requirements at each joint.

Since the end goal is to develop a biped capable of human-like walking, a published dataset of motion captured human gait was used for the initial dynamic simulation. Using Matlab toolboxes, the kinematic constraints on each joint ($\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$) were imposed on the dynamic

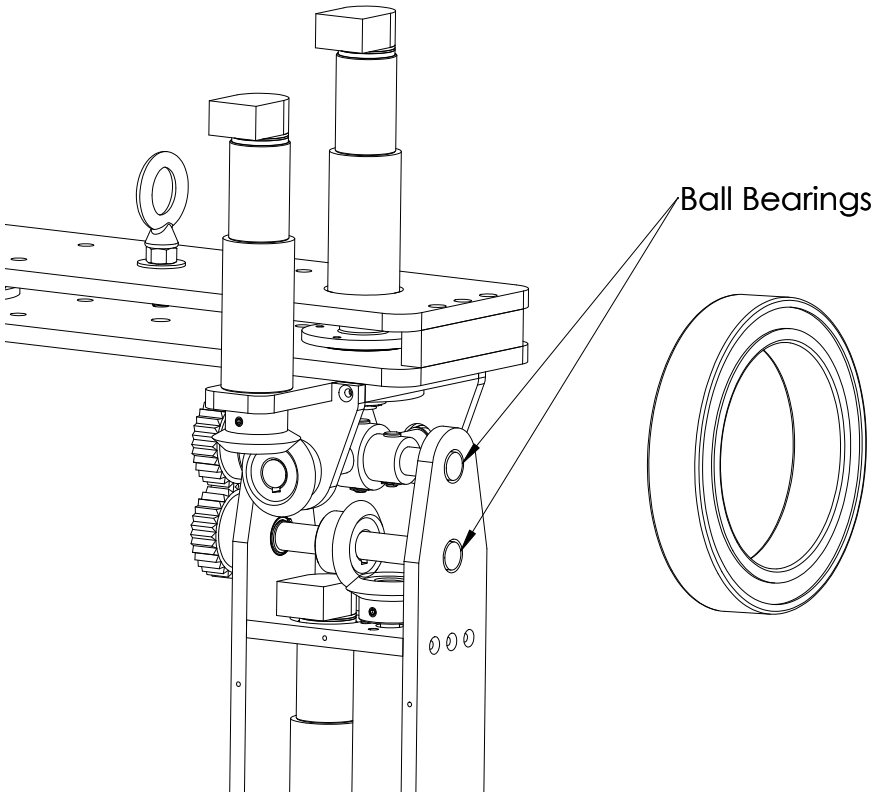


Figure 3.11: Enabling rotational motion of shafts using double shielded ball bearings.

model of the biped. A spring-damper contact model was used to generate an approximate ground reaction force experienced by the biped while walking. Finally, the initial joint torque and velocity demands under the estimated gait cycle were obtained through the dynamic simulations.

These initial design specifications formed the basis for drivetrain selection. The primary design goal of producing enough mechanical power for bipedal locomotion was addressed by appropriately sizing motors to the specifications. A combination of DC motors and precision gearheads were evaluated by analyzing the torque-speed, power, efficiency and thermal characteristics. The selection process revealed that a set of high and low power configuration of drivetrain components would meet the specifications while minimizing the overall weight of the system.

After the drivetrain configurations were established, the secondary design goal of mini-

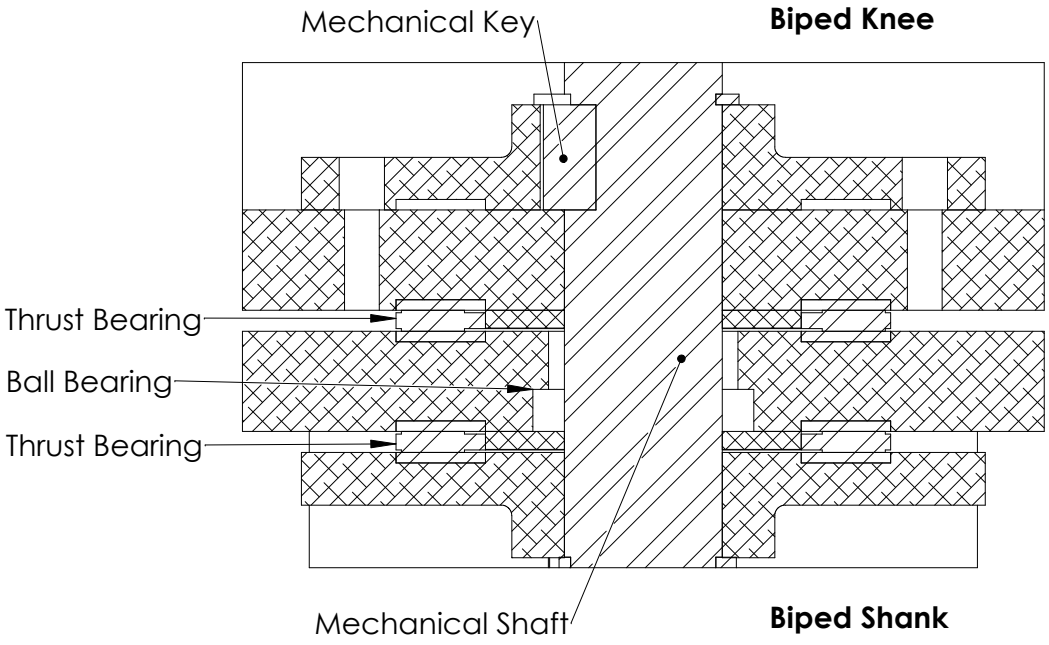


Figure 3.12: A combination of thrust and ball bearings used for yaw joints to support radial and axial loading.

mizing the overall weight and raising the COM were addressed through the mechanical design of the chassis. A rough guideline for dimensioning the lower body segments (e.g. thigh, shank) was obtained through anthropometric dimensioning and the desired height of the lower body. Aluminum 5052 was selected as the material to construct the chassis frame due to its light weight, fatigue strength and machineability. The power transmission system (composed of shafts, bearings and gears) connecting the actuators to the linkages were strategically designed to raise the COM and support the loading during gait.

Chapter 4

Toolchain Development[1]

This chapter introduces a rapid prototyping toolchain[2] developed to streamline the process of exporting a prototype design from a CAD software package to generate dynamic simulations with full 3D visualization. Starting from the initial prototype design and drivetrain specifications in the previous chapter, the design process outlined in Section 4.1 is used to improve the overall system performance. The toolchain provides an automated two-step process which begins by exporting key information from CAD (Section 4.2) and ends by regenerating the equivalent system in a simulation environment (Section 4.3). A case study is presented in Section 4.4 demonstrating the performance benefits of the proposed toolset.

The electromechanical design and development of multibody robotic systems is an iterative process, starting from a mechanical model in CAD software, transferring its parameters to a dynamic simulator for analysis and revising the design to improve the performance of the system. This process is repeated until the mechanical design achieves some desired goal. The iterative nature of the design and analysis process can often become time consuming and cumbersome. However, for high DOF multibody systems, the iterative approach is necessary as small changes in the mechanical design can have a significant impact on the overall dynamics of the system and the resulting system behaviour as well as implications for control design. Another popular approach is the use of optimization tools [99, 100] to determine the optimum design configuration. However, the resulting configuration may not be realizable with the available hardware components and must still be verified in simulation before hardware implementation. Ravichandran et al. used a hybrid

[1] A version of this chapter has been accepted for publication at the 2012 IEEE International Conference on Humanoid Robots [98]

[2] This work was completed in collaboration with Quanser Inc through the Natural Sciences and Engineering Research Council (NSERC)'s Industrial Postgraduate Scholarship Program.

evolutionary algorithm which optimizes discrete (e.g. available actuators) and continuous (e.g. link lengths) design variables to obtain the optimum design configuration which is realizable with commercial off-the-shelf parts [101]. This approach optimizes the design for *specific* tasks with predefined trajectories, which may be impossible to define for some multibody robotic systems with complex applications (i.e. bipedal robots with on-line gait generation).

4.1 Design Process

The initial prototype design process of the 14 DOF bipedal robot presented in Chapter 3 illustrates the need for a streamlined toolchain. Prior to obtaining the initial torque estimates from dynamics simulations, a basic bare-bones mechanical model of the bipedal robot was constructed in CAD (with $n = 14$ joints and $m = n + 1 = 15$ links). This initial mechanical model included arbitrary DC motor models from the manufacturer since the torque required was yet to be determined. After executing the dynamic simulations using Kajita's toolbox for the first time, the torque estimates were used to revise the mechanical design in CAD with more appropriate motors. However, this design revision altered the robot dynamics (e.g. increased mass due to larger motors) which effects the joint torques required to support the updated mechanical model during the estimated gait cycle. The design process of executing the dynamic simulations, obtaining updated torque estimates and revising the mechanical model with more appropriate motors was repeated several times. The process ended once a suitable motor configuration capable of producing the estimated torque required its own mechanical model was found. The frequent and incremental design revisions required moving back and forth between the CAD software package and the dynamic simulation environment. Each design revision required transferring the updated kinematic and dynamic parameters of the links back to the simulation environment prior to executing the simulation.

There exists a wide variety of dynamic simulation environments with a range of capabilities [102, 103, 104, 105, 106, 107]. These environments provide a feature-complete package, which combine an underlying dynamics engine with an interface for visualizing the simulations. The underlying computation engines obtain the complete equations of motion for the system described by its kinematic/dynamic parameters using techniques including, but not limited to, Lagrange multipliers [108], Kane's method [109] and port-based modeling [110]. The equations of motion are integrated to obtain the system state at each time step. Most simulators provide support for importing common Virtual Reality Markup Language (VRML) or Standard Tessellation Library (STL) files generated by CAD tools for visu-

alization [102, 103]. However, only a few provide direct compatibility with CAD tools to import kinematic/dynamic parameters for simulation.

Open Architecture Humanoid Robotics Platform (OpenHRP) [104] is a commonly used dynamic simulation environment for humanoid robots that supports importing VRML files. However, kinematic and dynamic parameters of the robot are specified as plain text, making it cumbersome for rapid iterations.

MapleSim [111] is a multibody simulation package that provides limited functionality to communicate with CAD applications through its MapleSimConnector toolbox. This toolbox uses the underlying Maple engine with command-line access for retrieving the parameters of each link. However, the CAD application has to be actively running in the background and it is left to the user to generate Maple worksheets for batch importing of the parameters to update the link parameters in MapleSim through its API.

SimMechanics [112] provides mechanical import functionality to generate Extensible Markup Language (XML) files containing link parameters directly from CAD along with the corresponding STL files for visualization. However, there are limitations on how joint constraints are defined in CAD to successfully generate an equivalent model in Simulink. Another drawback of this approach is that the visualization generated during simulations significantly impacts the simulation speed.

4.2 CAD Export

An add-in was developed for CAD software package SolidWorks to export a multibody system for dynamic simulations in Simulink with realtime visualization. Consider a standard multibody system with n joints and $n+1$ links. The links are numbered from 0 (base) to n and each $joint_i$ connects $link_{i-1}$ to $link_i$. The mechanical design of each link is represented by its own CAD assembly (or part) file. The coordinate system at the origin of this CAD file is treated as the local coordinate system xyz_i rigidly attached to the link i. The top-most CAD assembly (referred to as the *master assembly*) contains all $n+1$ links as subassembly files connected and constrained by the mechanical relationship which defines the behaviour of each joint n. For example, a revolute joint connecting two link subassemblies is defined by the appropriate constraints (e.g. concentric relationship).

The add-in for SolidWorks uses this master assembly file as a starting point to export the multibody system for dynamic simulations. Once installed, a new tab is added directly in SolidWorks (as shown in Figure 4.1) presenting the user with several export options. The initial export process prompts the user with a flat list of all subassemblies in the current

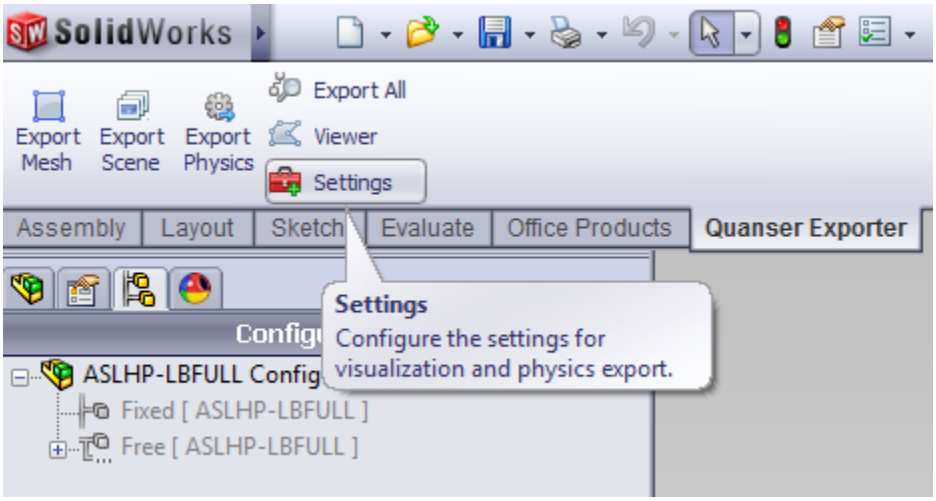


Figure 4.1: The exporter add-in for SolidWorks (named Quanser Exporter in the SolidWorks tab) to capture relevant physical and visualization data.

file via a *Model Organizer* interface (shown in Figure 4.2). Each subassembly is treated as a node that can be ordered (through a drag-and-drop interface) to form the desired kinematic tree/chain hierarchy.

4.2.1 Physics Export

Each link subassembly is parsed in the order defined in the *Model Organizer* window to extract key kinematic/dynamic parameters. All numerical values (e.g. distance between links) are extracted directly using CAD tools. Assuming that the system is in the home configuration at the time of export, the relative frame transformations T_i^{i-1} between subsequent frames are extracted in accordance with the kinematic hierarchy defined in the *Model Organizer*. The absolute frame transformations T_i^0 are also computed with respect to the base link (if the base link is fixed in the master assembly file). For floating base systems (i.e. base link has 6 DOF), the absolute frame transformations T_i^W are taken with respect to the coordinate system of the master assembly file (world frame). The mass and dynamic properties are extracted with internal CAD tools in the local coordinate system of the link subassembly file. The mass, distance to the center-of-mass (COM) and inertia tensor of the link (at the COM position) are extracted.

The add-in captures these key parameters and generates a structured XML file. An XML node is generated for each link containing the extracted kinematic/dynamic parameters

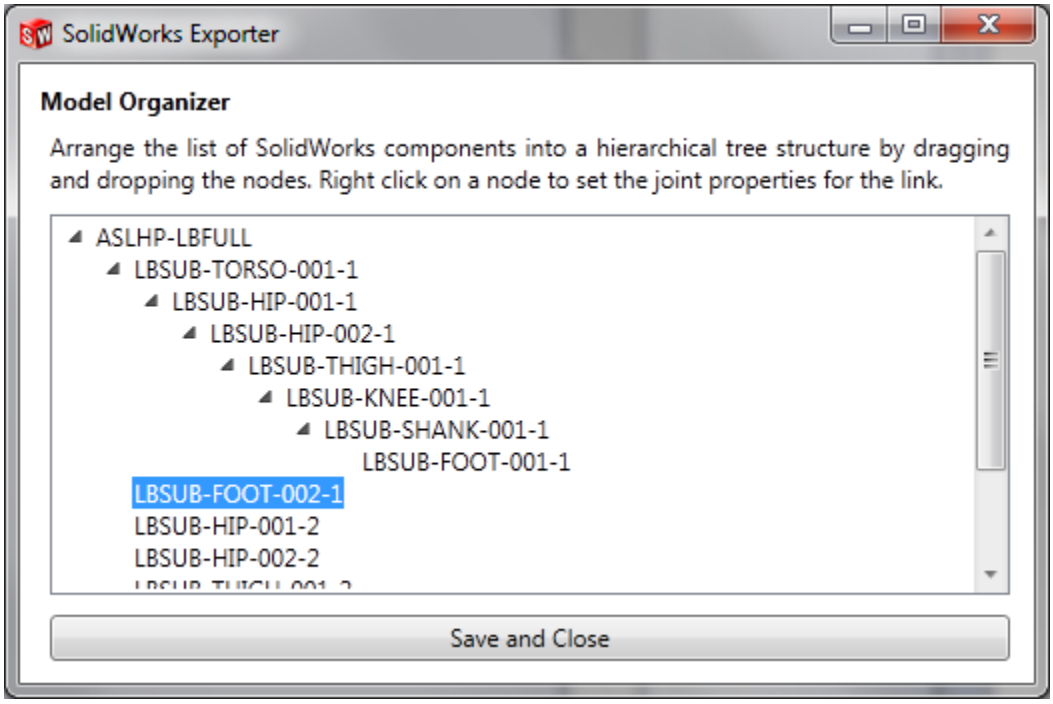


Figure 4.2: The *Model Organizer* window used to define the kinematic hierarchy of links during initial export.

(as shown in Figure 4.3). Furthermore, each XML link node is organized in accordance with the kinematic hierarchy of the system. As a result, the add-in captures the entire physical description of the multibody system in a portable, language-independent file.

4.2.2 Mesh/Scene Export

In addition to exporting the physical description of the model, the exporter add-in also generates files for visualization. These files can be used in conjunction with dynamic simulations to provide the user with visual feedback of the system under control. Some dynamic simulators (SimMechanics, MapleSim) currently allow users to specify a VRML file for each component of the multibody system. While SolidWorks currently has some support to export VRML files for each link subassembly individually, there are no options to generate the 3D meshes in X3D format (successor to VRML).

The toolchain provides automatic batch generation of full 3D meshes in the X3D format for each link subassembly. Once the export process has been initiated, the meshes are

```
- <MassProperties>
    <Mass>0.534887028663829</Mass>
  - <COM>
      <X>-3.9E-07</X>
      <Y>0.04741578</Y>
      <Z>-0.00980513</Z>
    </COM>
  - <Inertia Format="Row Major" Columns="3" Rows="3">
      <Ixx>0.001186264679004351</Ixx>
      <Iyy>0.00070094139771874721</Iyy>
      <Izz>0.0014829006492190239</Izz>
      <Ixy>-7.9742722453230943E-05</Ixy>
      <Izx>3.6541669161088704E-05</Izx>
      <Iyz>-0.00014857992698925752</Iyz>
    </Inertia>
    <Volume>0.00016279636319438745</Volume>
    <Area>0.085515263680874087</Area>
  </MassProperties>
- <MaterialProperties>
  - <Colour>
      <Red>0.753</Red>
      <Green>0.753</Green>
      <Blue>0.753</Blue>
    </Colour>
```

Figure 4.3: Example of kinematic and dynamic parameter extraction straight from CAD model stored in the exported XML file.

extracted for each link subassembly in the master assembly file directly from the CAD layout. Figure 4.4 shows the X3D mesh generated from a link subassembly in CAD. Each mesh is aligned with the local coordinate system of the link such that there is a direct mapping between the visualization and the physical model.

The overall mesh generation process is optimized for complex multibody systems. A typical CAD subassembly representing each link may contain hundreds of smaller CAD part/assembly files. During the mesh generation process the add-in creates a new flattened version of each link subassembly and discards any visual details not visible on the surface. The surfaces of this flattened model are tessellated to generate the mesh file for each link. As a result, the mesh files are light weight and this helps speed up the rendering process.

The output mesh files are stored in the same directory as the XML file containing the physical description of the system. Furthermore, each link XML node is also updated with a complete path to its corresponding mesh file alongside the kinematic/dynamic

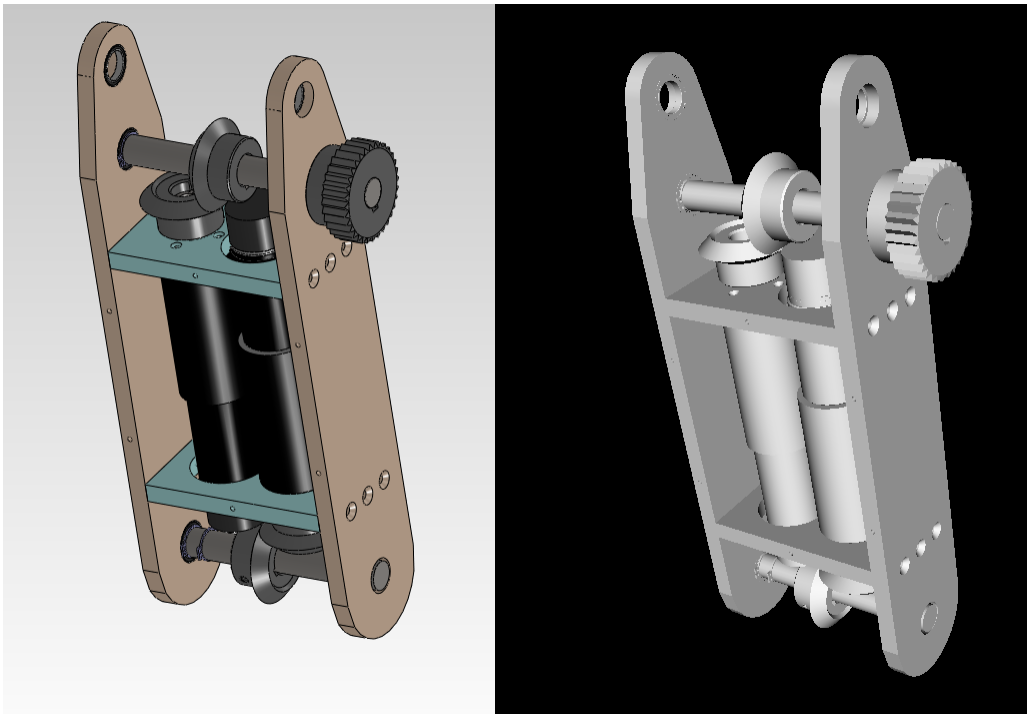


Figure 4.4: Equivalent X3D mesh file (shown on the right) generated directly from CAD layout (shown on the left) for a single link subassembly.

parameters. The add-in also generates a *scene* file from the CAD master assembly. This is a parallel XML file which contains a complete visual description of the system from CAD. The scene file contains a list of all links alongside their corresponding mesh files. These meshes are then organized in accordance with the kinematic tree/chain hierarchy provided by the user in the *Model Organizer*.

As a result, the scene file recreates the visual layout of the complete multibody system from the CAD master assembly file. This file encapsulates the full multibody system in a format which is compatible with the 3D visualization blocks that are packaged with the QUARC toolbox. In addition to the model itself, additional supplementary mesh files (e.g. ground plane) are imported into the scene for visualizing the environment.

4.2.3 CAD Update

With minimal user input, the add-in captures a complete physical and visual description of the system in its *current state*. However, the main benefit of this toolchain becomes apparent after the initial export. Once the user defines the structure of the system (kinematic hierarchy and joint definitions) through the *Model Organizer*, it is stored in memory for later use. Subsequent updates to the CAD model can be exported with a single click if the overall structure of the links and joint definitions do not change.

This allows the user to export a model for simulation from CAD, analyze the behaviour of the system under control, tweak the mechanical design and immediately re-export the revised model for simulation. For example, increasing the length of a link may alter its dynamic properties while the overall kinematic structure remains the same. The revised CAD model with increased link length can be exported with a single click. The export process simply recalls the structure of the system and regenerates the XML files with updated kinematic/dynamic properties. The add-in also allows the user to regenerate only the mesh file for the updated link to reflect the changes in CAD. This process makes it fast and easy for rapid iterations during the design phase.

4.3 Model Generation

The XML files exported by the add-in encapsulate the relevant CAD model information into structured and portable files. One of the key advantages of this approach is that the information in these files can easily be parsed to generate an equivalent model in most dynamic simulators. A model generation counterpart was developed for the Matlab/Simulink environment which uses SimMechanics for multibody dynamic simulation and Quanser's QUARC toolbox for 3D visualization. This provides a semi-automated toolchain for designing robotics and/or mechatronic systems.

The model generation approach provides several Matlab functions, scripts and libraries to parse the files exported by the SolidWorks add-in and generate the equivalent mechanical system in Simulink. The generated Simulink model contains (a) SimMechanics blocks with the link kinematic and dynamic parameters from CAD and (b) visualization blocks from QUARC libraries with the link meshes and scene file. The generated physics and visualization counterpart is pre-populated with CAD data and connected according to the kinematic hierarchy of the system defined in the *Model Organizer*.

The model generation process is initiated by calling a function in the Matlab terminal inside the CAD export folder. By default, this approach generates a physical subsystem (i.e. the

plant) for forward dynamics simulation, whose output drives a generated visualization subsystem. The user can also generate an inverse dynamics model through the command line.

4.3.1 Physical Model

The physical model generation process parses the XML output files in the export folder to create a CAD-equivalent system in Simulink. In order to streamline the process, a library of masked link subsystems representing common link configurations (shown in Figure 4.5) was developed. Each of these subsystems contains a combination of SimMechanics joint, body, actuator and sensor blocks to represent a combination of $joint_i$ and $link_i$ (shown in Figure 4.6).

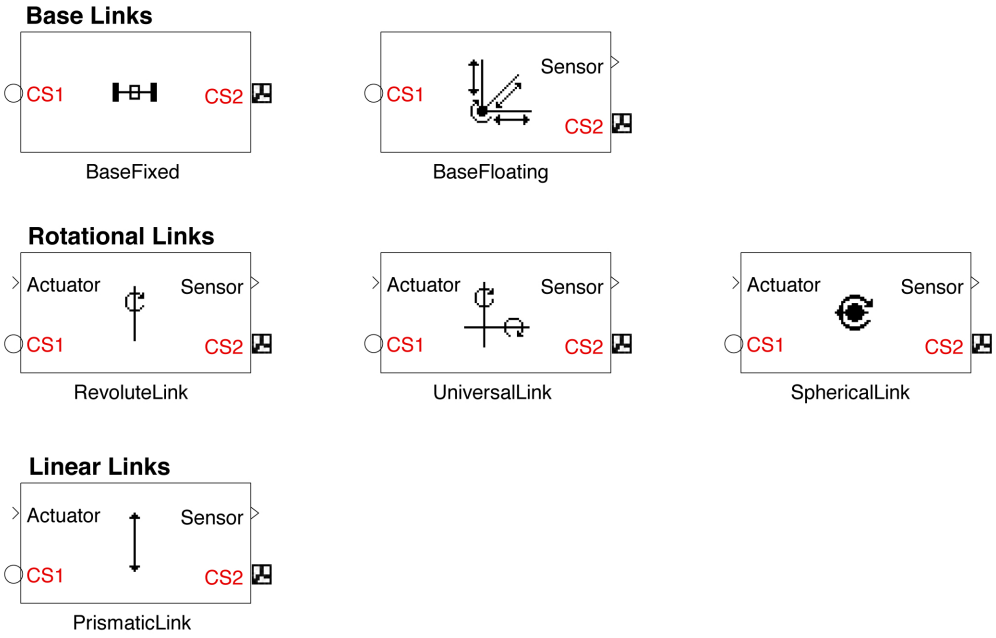


Figure 4.5: The standard link subsystems for physical model generation with prepopulated kinematic and dynamic parameters from CAD.

The input to each link subsystem (**CS1**) is a connection from $link_{i-1}$ and the driving signal for the joint. The output of the link subsystem is its local coordinate system (**CS2**) and the joint sensor signals. The joint actuator/sensor signals are set depending on the type

of dynamic simulation. By default the model generation process configures the physical model in forward dynamics mode so the joint actuator port is connected to the force/-torque command input for $joint_i$ and joint angle/displacement is measured at the output. Alternatively, for the inverse dynamics simulation, the joint acceleration is set as the input, while the joint force/torque is the output. The output coordinate system of the $joint_i$ block is connected to a SimMechanics body block representing $link_i$. The masked parameters are preconfigured to set the local dynamic properties (e.g. COM position, inertia tensor) and relative frame transformation appropriately.

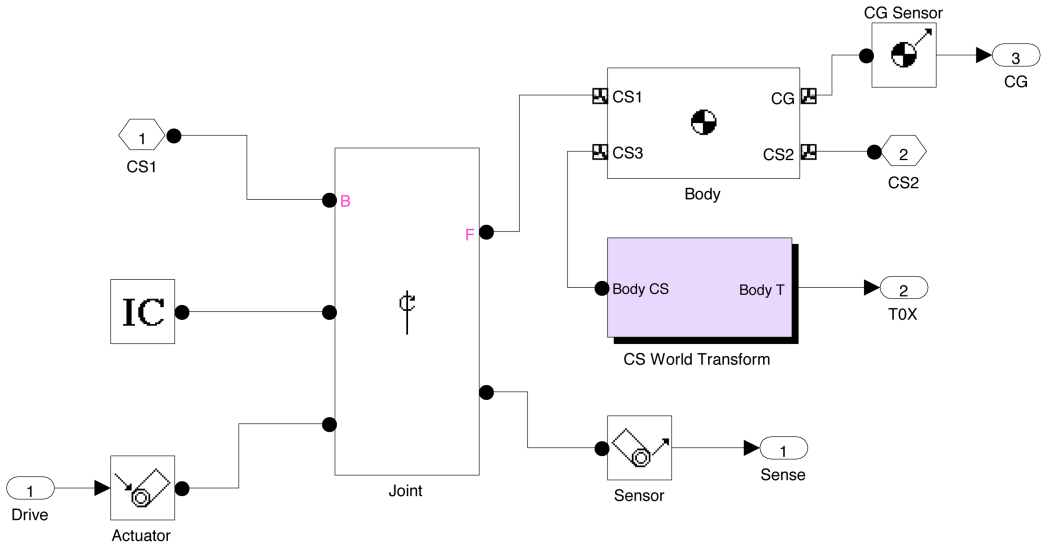


Figure 4.6: SimMechanics blocks used to compose each CAD-equivalent link subassembly in Simulink

Each link subassembly from CAD is recreated with a link subsystem from the library depending on the link type. The masked parameters are populated with the kinematic and dynamic parameters from the corresponding XML link node. The overall hierarchy of links is parsed and each link subsystem is connected accordingly. The model generation process also automatically handles the signal routing from the input/output ports of the physical model. In the default forward dynamics configuration, the $n \times 1$ torque/force vector is connected to all joint actuator blocks and the output signals from the sensors are also routed accordingly. For the inverse dynamics case, the vector of joint positions, velocities and accelerations are connected to the joint actuator blocks and the sensors are preconfigured to output the resulting torque/force vector.

In addition to the link subsystem, a ground block representing the coordinate frame of

the CAD master assembly is connected to the base $link_0$. The entire process of creating an equivalent model for dynamic simulation is automated. The user simply calls the appropriate function and the generated physical model subsystem is placed in a new (or existing) Simulink diagram.

4.3.2 Visualization Model

The visualization model generation process generates a single subsystem containing blocks to initialize and drive the scene file generated from CAD. The mesh for each link in the kinematic chain is driven by the output of the generated physical model (forward dynamics subsystem) so that there is a 1:1 mapping between the plant and what is being rendered by the visualization. When a simulation is running, an external 3D viewer application (part of the QUARC toolbox) is used to render the scene in real-time using OpenGL (CAD-equivalent visualization scene file shown in Figure 4.7).

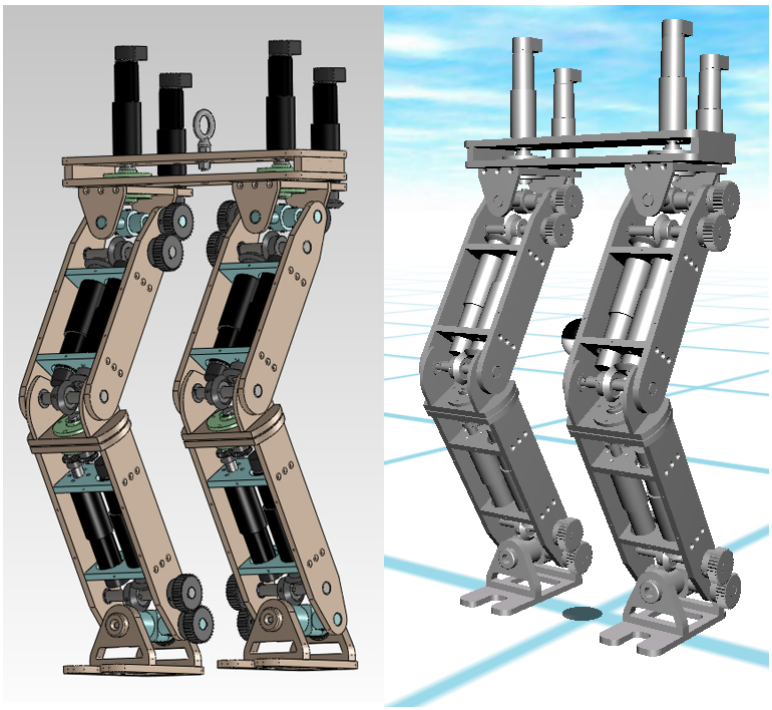


Figure 4.7: CAD master assembly shown on the left is used to automatically generate the meshes and scene file to recreate the visualization shown on the right.

This approach allows the user to receive immediate visual feedback of the CAD model under the influence of control. This also allows the user to make changes to the mechanical design from visual observations (e.g. quick changes to improve joint limits).

4.3.3 Model Update

The exporter add-in makes it possible to make changes to the mechanical design in CAD and generate the new kinematic and dynamic parameters immediately. After the initial model generation from CAD data, these new changes can be reflected back into the dynamic simulation by simply calling the update function. The update functionality also allows the user to specify the path to the previously generated physical model. The masked parameters on each link subsystem inside this model are simply updated with the new changes while the signal routing and everything else is left intact. The updated mesh files are loaded by the QUARC 3D viewer during the next simulation run.

This streamlines the iterative design process by allowing a user to export a system from the add-in and generate the CAD-equivalent physical/visual model in Simulink. The behaviour of the system can then be analyzed in dynamic simulations to revise the CAD design. Once the new changes are applied, the revised parameters are easily exported back into dynamic simulations for further analysis.

4.4 Case Study

This toolchain was used to improve the initial design of the 14 DOF lower body humanoid robot[3] discussed in Chapter 3. The toolchain was used to quickly analyze the effects of design revisions (e.g. compare motor positioning) in simulation prior to manufacturing. The complete initial export and model generation process takes about two minutes for the 14 DOF bipedal robot. Since the kinematic hierarchy and joint information is stored during the initial export process, subsequent exports take around thirty seconds to reflect the design revisions from CAD back into the dynamic simulator.

[3] A video demonstration of the toolchain being used is available online at:
`https://ece.uwaterloo.ca/~dkulic/videos/Humanoids2012-480p.mp4`

4.4.1 Dynamic Simulations

The proposed toolchain was used during the design phase to estimate the torques at each joints for appropriate motor sizing. Note that the mechanical design in CAD includes the selected motor models from the manufacturer. Changes to the mechanical design such as motor positioning and material of the links can significantly alter the torques required at some joints. In these situations it is useful to make incremental changes to analyze their impact on the system performance and immediately use this knowledge to tweak the design.

A common motion for humanoid robots is the bending of the knee and hip joints while a foot is swinging over during the gait cycle. During this phase, the motors at the hip joint must carry the overall weight of the leg below it. The choice of actuators on the leg plays a crucial role in its overall weight and as a result, also plays an important role in the torques required at these joints. Similarly, when the swing foot comes into contact with the ground, the knee joint absorbs a significant amount of torque so the motors must be sized appropriately.

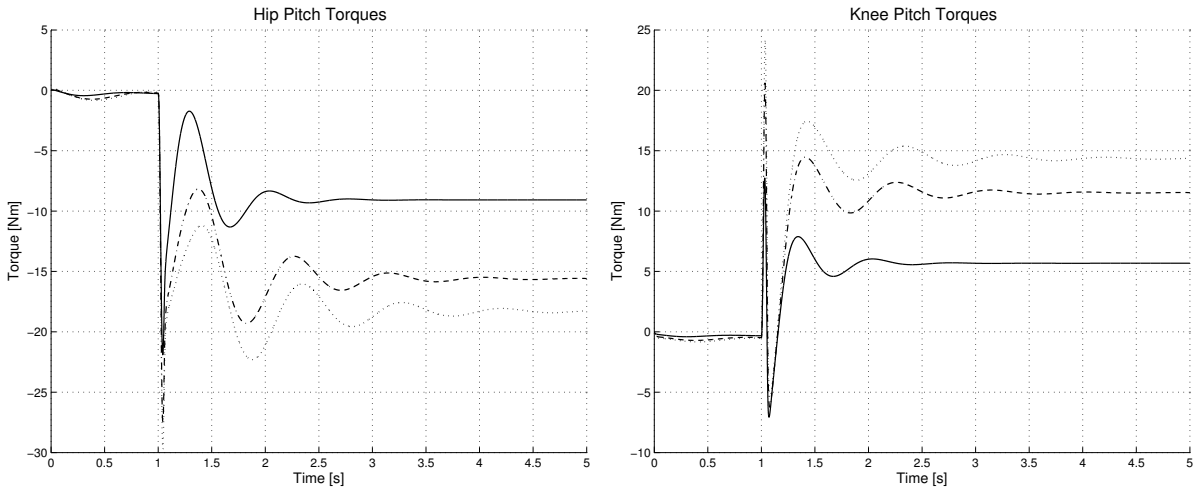


Figure 4.8: Hip and knee joint torque requirements while a leg is raised for different sets of motors at the joints.

By using the toolchain, the user can maintain multiple configurations of the same mechanical design in CAD and export them for direct comparison with the dynamic simulator. Since the model is already defined, exporting each configuration takes less than a minute.

The parameters of these configuration files can be used along with the model update feature to simulate each configuration and compare the results. In the case of bending the knee and hip joints, there may be several choices of motors which alter the system performance (due to large masses further down the leg). The torque profiles of each joint can be analyzed in the simulation (as shown in Figure 4.8) to select the most desirable configuration. The dotted and solid lines represent the heaviest and lightest motors, respectively. The dashed line represents the motor set with medium weight (between the heavy and light motors). The toolchain allows for fast incremental changes to revise the mechanical design and compare the torque requirements.

4.4.2 Visualization

Additional meshes can also be added to the scene to serve as visual aids. For example, coordinate systems can be visualized during simulation by attaching arrow meshes to each link. Alternatively, common Simulink blocks can be used to determine if a particular joint is out of its limits and use the resulting signal to drive the colour of the mesh file (i.e. turn a link *red* if the joint is out of its limits). A particularly useful visual aid for floating base multibody systems is the ability to see where the COM of the system is during simulation. An additional mesh can be added to the scene driven by the COM calculation from forward kinematics at each time step. The use of this visual aid is shown in Figure 4.9.

Table 4.1: 60s Dynamic Simulation Runtime

Configuration	**Runtime** (s)
No Visualization	47.50
SimMechanics Visualization	367
Proposed Toolset Visualization	49

By having the visualization rendered in an external application, the proposed toolchain is capable of running simulations much faster than the built-in 3D viewing capabilities of SimMechanics. This is due to the fact that SimMechanics is unable to parallelize the multibody simulations and visualization. This increase in runtime speed is especially useful during the design phase where rapid iterations are common. The runtime for a 60s dynamic simulation[4] at 1KHz *with* visualization is compared in Table 4.1. If the viewer is also being

[4]The simulations were executed on a standard PC available at the time of writing (2.4GHz Intel Core 2 Duo, 4 GB RAM).

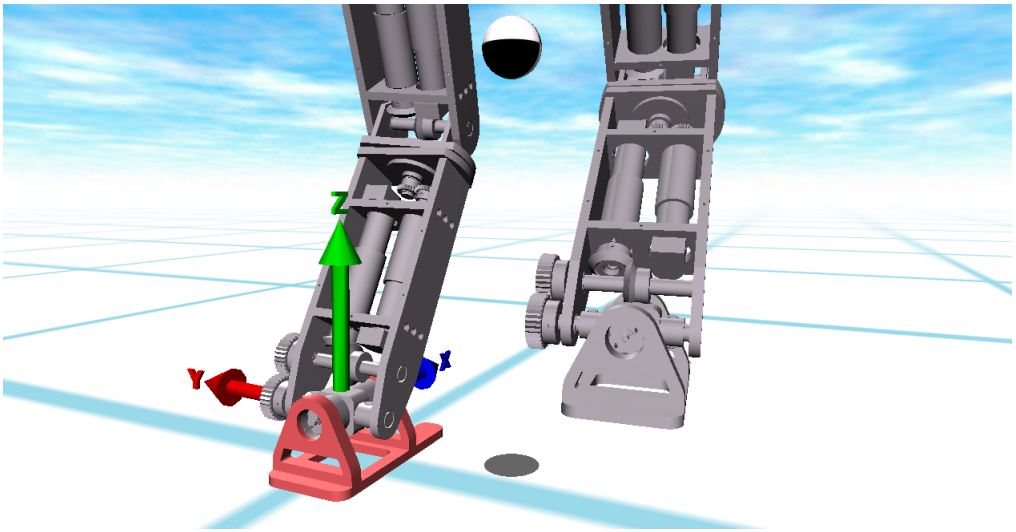


Figure 4.9: Realtime visualization during simulations allows the user to get immediate visual feedback on important information like COM position.

used to render the 3D mesh for each link through supplied VRML files, it takes much longer. In contrast, the proposed toolchain takes a fraction of the time for the exact same system to be simulated. The simultaneous visualization using the QUARC visualization toolset has little or no impact on the simulation speed. In fact, running the same dynamic simulation without the QUARC visualization toolset only reduces the overall runtime slightly.

4.5 Summary

A streamlined toolchain was developed to enable rapid prototype development of complex multibody systems. The design process for these systems starts from a mechanical design in CAD, transferring the kinematic/dynamic parameters to a dynamic simulation environment for analysis and revising the design to improve its performance. While a wide range of dynamic simulators already exist, very few of them provide the functionality to directly import the kinematic/dynamic parameters from CAD. The toolchain provides a semi-automated two-step process for extracting the relevant parameters from CAD and automatically generating the equivalent model in a dynamic simulator.

The CAD export process is activated through a native plug-in developed for SolidWorks. The plug-in processes each link subassembly file to extract key physical parameters (kine-

matic/dynamics) and tessellates the surface of each link to generate a 3D mesh for visualization. During the initial export, the user is presented with a simple drag-and-drop interface to specify the kinematic hierarchy of links and joint definitions for the system. The extracted physical and visualization data are stored as XML files in subdirectory for subsequent model generation.

By storing the extracted data in the universal XML format, routines can be programmed to regenerate the CAD-equivalent model for almost any dynamic simulator. A model generation counterpart was developed for the MATLAB/Simulink environment. Calling a single function in MATLAB automatically parses the exported data and generates full dynamic simulations in Simulink using SimMechanics and 3D visualization using the QUARC toolbox. After the initial model generation, subsequent exports only update the physical model parameters and/or visualization blocks enabling faster design iterations.

The toolchain was used extensively during the design and development of the 14 DOF biped, demonstrating its usefulness in designing a physical robot. Through the use of visualization aids (i.e. for COM and ZMP position), it was very easy to analyze the behaviour of the initial electromechanical design (from Chapter 3) under control action. The forward dynamics and 3D visualization generation proved to be a tremendous benefit during the development of the walking control strategy discussed in the following chapter.

Chapter 5

3D Foot Placement Estimator and Gait Generation[1]

This chapter extends the Foot Placement Estimator (FPE) algorithm for 3D bipedal robots. Section 5.1 presents the method of extending the existing 2D theory (reviewed in Section 2.2.5) to 3D movement. The proposed algorithm selects a 2D plane in the chosen direction of motion and generates trajectories to produce a forward momentum along the plane. A whole-body motion control framework coupled with a finite state machine is used to track the generated trajectories and form complete gait cycles. The toolchain described in the previous chapter is used to generate dynamic simulations for the 14 DOF bipedal robot and the proposed extension to 3D is validated (Section 5.2).

The remainder of this thesis assumes a 3D bipedal robot with n actuated degrees of freedom (DOF) and $n + 6$ generalized coordinates defined by (3.2). The biped's local coordinate system $oxyz$ rigidly attached to the floating base is shown in Figure 5.1. The xz and yz planes form the biped's *local* sagittal and frontal planes, respectively.

5.1 FPE Extension to 3D

In order to extend the FPE approach to the 3D case, the concept of generating complete gait cycles described in Section 2.3.6 is revisited. The primary goal of the first three states in each step cycle (**PUSH**, **LIFT** and **SWING**) is to force the biped into an unstable

[1] A version of this chapter has been submitted to the 2013 IEEE International Conference on Robotics and Automation

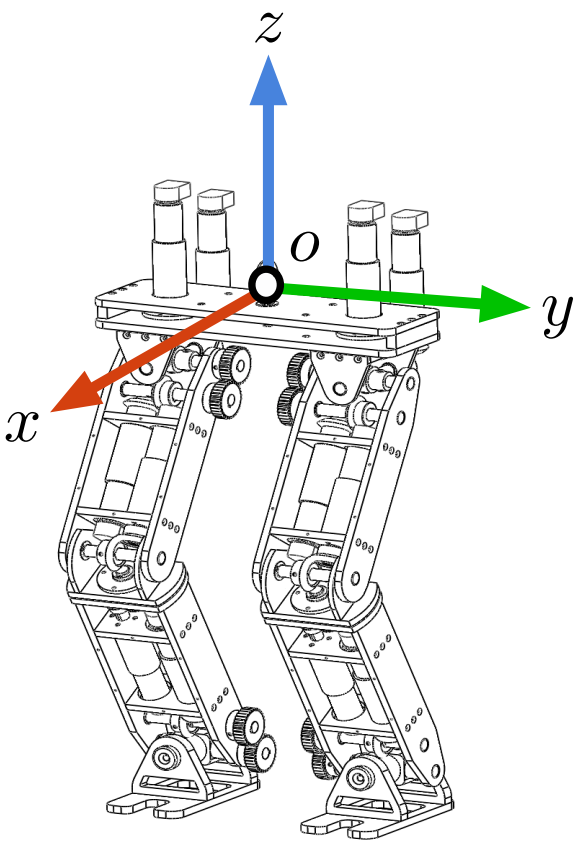


Figure 5.1: Local coordinate system for the 3D 14 DOF bipedal robot.

configuration so that the FPE algorithm can be used to regain stability in the terminal state (**DROP**).

To extend the 2D algorithm to the general 3D case, we begin by selecting a suitable plane in 3D space as the sagittal plane for motion. The frontal plane is perpendicular to the sagittal plane and the ground. In the proposed approach, the goal of each step cycle is to control the motion of a 3D bipedal robot to generate a forward moving momentum *along* the selected sagittal plane. Upon entering the terminal state, the FPE equation (2.11) is solved on the selected plane to determine the swing foot placement and ultimately regain stability. Unlike the 2D case, consider a 3D bipedal robot with finite foot length and width rather than a biped with point feet as demonstrated in [7]. The larger size of the region of support increases robustness to the approximation errors of simplifying the 3D dynamics to a 2D plane.

5.1.1 Sagittal Plane

To select an appropriate sagittal plane for a 3D bipedal robot, a vertical plane which lies between the current position of the stance foot and the desired direction of motion is chosen. For a 3D biped walking in a forward motion, this plane is chosen as the vertical plane passing through the midpoint between the hips and parallel to the direction of forward progress (as shown in Figure 5.2). For side-stepping motion, the frontal plane through the stance foot in the direction of the side step is chosen as the plane of motion (as shown in Figure 5.3).

The motion of the biped is controlled based on the selected plane for the duration of the step cycle. During gait initiation, the lines from the COM to the contact points are of length L, and the leg separation angle is β (similar to the planar case). If the motion of the biped is constrained along this plane, the FPE angle ϕ can be used to determine foot placement to regain stability. The parameters required to solve the FPE equation are projected onto the selected plane (additional details are provided in Section 5.1.5).

Upon impact, the angle ϕ converges to $\beta/2$ and a new sagittal plane can be selected for the subsequent step cycle prior to the swing leg entering the **PUSH** state. Once selected,

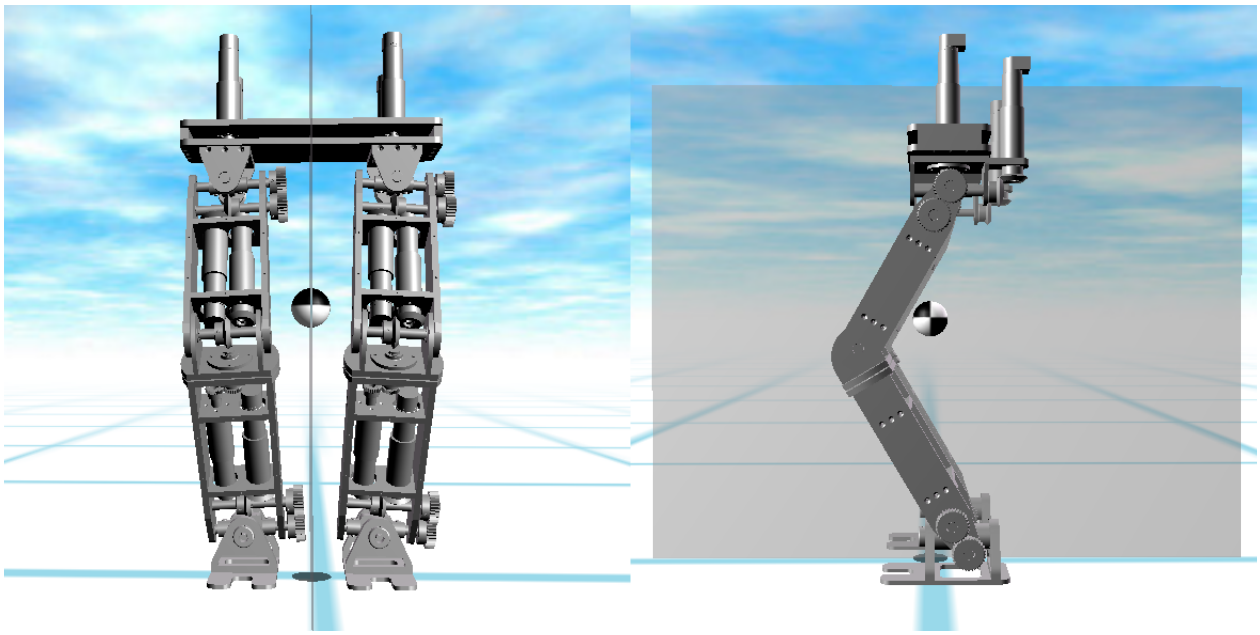


Figure 5.2: Sagittal plane selection (shown as translucent gray) for forward walking.

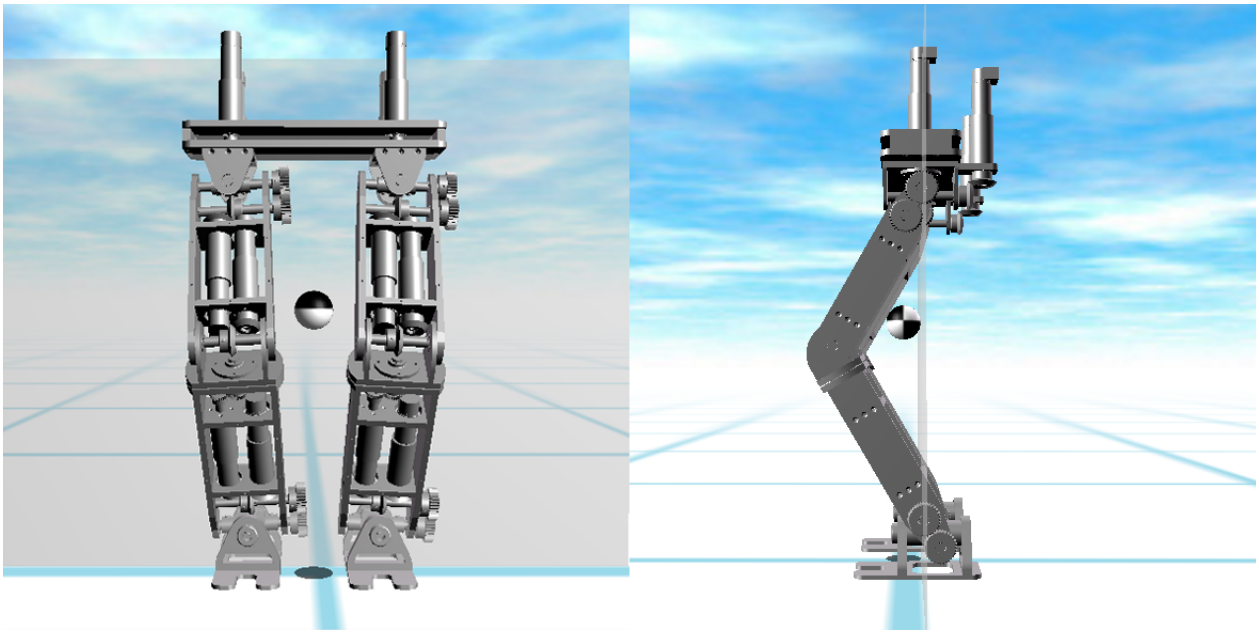


Figure 5.3: Sagittal plane selection (shown as translucent gray) for side-stepping.

the stance foot is rotated for alignment and swing leg trajectories can be generated along the plane. By selecting a plane between the current and desired directions of motion, this approach can achieve turning with each step.

5.1.2 Trajectory Generation

Once the sagittal plane has been identified at the beginning of each step cycle, appropriate task space trajectories must be generated for the COM ($\mathbf{x_{COM}}$) and the swing foot ($\mathbf{x_{SWING}}$). In the 2D case, the main goal of the initial states **PUSH**, **LIFT** and **SWING** was to achieve enough forward motion to destabilize the biped. In the 3D case, the robot must also remain stable in the off-sagittal plane while achieving the desired sagittal plane motion. If the ZMP leaves the region of support formed by the stance foot as the swing foot is lifted, the biped begins to fall in the off-sagittal plane and the solution to the 2D FPE equation is insufficient to maintain stable gait. To ensure both forward progress and off-sagittal plane stability, the generated trajectories for $\mathbf{x_{COM}}$ are shown in Figure 5.4.

PUSH: $\mathbf{x_{COM}}$ is moved above the leading stance foot to maintain stability in the off-

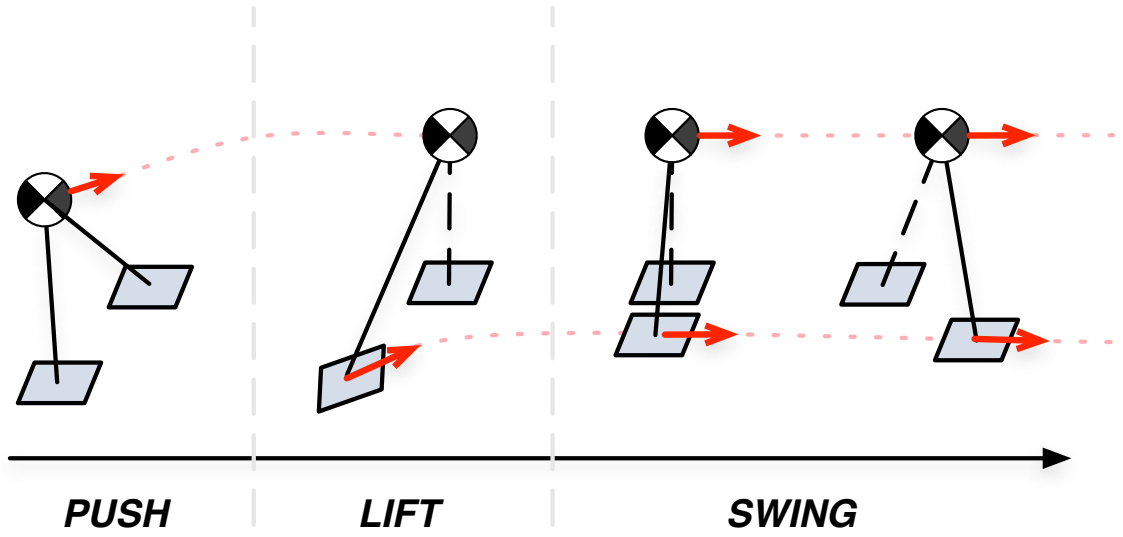


Figure 5.4: Trajectory for $\mathbf{x_{COM}}$ to ensure forward progress and off-sagittal stability.

sagittal and sagittal planes.

LIFT: $\mathbf{x_{COM}}$ is held at its current location (above stance foot) while the swing foot is lifted from the ground to achieve sufficient clearance.

SWING: $\mathbf{x_{COM}}$ is held in place until the swing foot is aligned with the stance foot in the off-sagittal plane. At this point the $\mathbf{x_{COM}}$ is deliberately pushed outside the region of support in the sagittal plane direction.

A similar approach is used to generate trajectories for $\mathbf{x_{SWING}}$ to achieve the desired behaviour of generating enough momentum to destabilize the biped in the sagittal plane while maintaining stability in the off-sagittal plane. Trajectories for $\mathbf{x_{SWING}}$ (illustrated in Figure 5.5) are always computed to align with the sagittal plane formed by the stance foot at the start of the step cycle. This ensures that the solution to the 2D FPE equation remains valid as the **DROP** state is entered.

PUSH: $\mathbf{x_{SWING}}$ is held in place as the $\mathbf{x_{COM}}$ trajectory is tracked.

LIFT: $\mathbf{x_{SWING}}$ follows a ramped trajectory to simultaneously raise the foot off the ground and move it forward in the sagittal plane.

SWING: $\mathbf{x_{SWING}}$ follows a straight line trajectory at a specific ground clearance (shown

as h_{LIFT} on Figure 5.5) until it reaches the FPE angle ϕ or the biped is unstable.

The ramp trajectory used to raise the swing foot during **LIFT** should be parameterized in terms of the velocity of the FPE point so that this state transitions faster in the event of larger disturbances (since the biped would have a shorter amount of time to swing the foot over and catch itself).

Depending on the supervisory control mode (i.e. **WALK** or **STAND**), the swing leg trajectory can be adjusted to implicitly achieve a desired goal. During **WALK** mode, the swing foot trajectory tracks a point on the ground slightly behind the FPE point. This under stepping behaviour results in the biped having enough forward moving momentum when the swing foot comes in contact with the ground such that the biped is unstable. As a result, the FPE point is continuously moving forward causing the state machine to transition into the opposing foot's **LIFT** state upon contact. In the **STAND** mode, the swing foot trajectory is adjusted to overstep the FPE point so that the biped comes to a stop following this step.

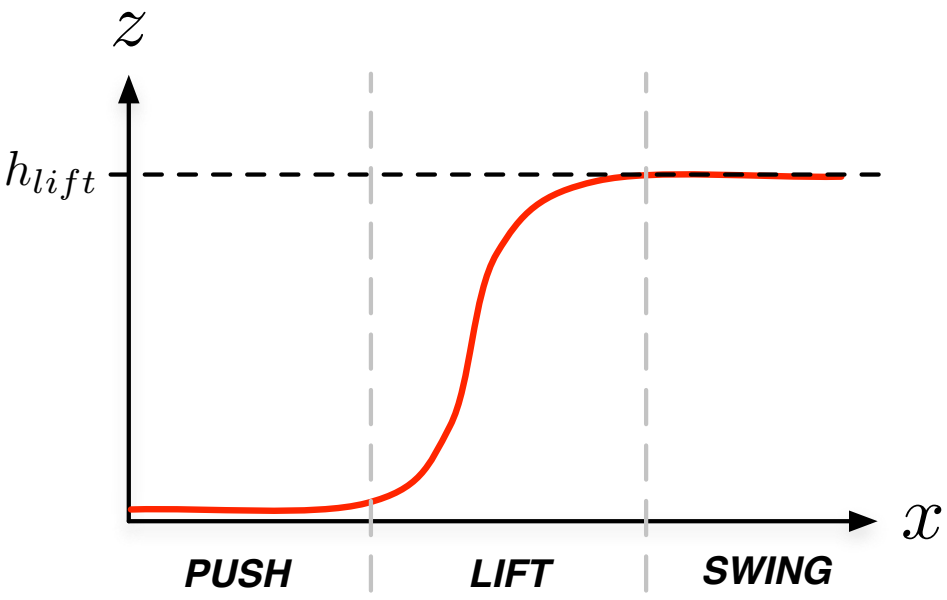


Figure 5.5: Trajectory for $\mathbf{x_{SWING}}$ along the selected (sagittal) xz-plane.

5.1.3 Control Strategy

A hybrid control strategy is used to simultaneously maintain stability in the off-sagittal plane, achieve sufficient forward momentum along a selected sagittal plane and ultimately track the FPE location to regain stability by taking a step. Similar to the approach presented in [7], this approach uses a state machine to transition through the step sequence with each state having a local controller.

During the initial states of the step cycle, whole body motion control is used to track the $\mathbf{x_{COM}}$ and $\mathbf{x_{SWING}}$ trajectories described in Section 5.1.2. To generate the corresponding joint level trajectories, the Jacobian matrix is used to map between the task space and the joint space velocities:

$$\mathbf{J} = \begin{bmatrix} \partial \mathbf{q_{act}} & \partial \mathbf{x_{base}} \end{bmatrix}_{m \times (n+6)} \tag{5.1}$$

A prioritized task space control scheme is used to generate joint level trajectories which simultaneously achieve state goals while satisfying the highest priority constraint (i.e. holding the $\mathbf{x_{COM}}$ position). The state-dependent joint level trajectories can be computed by projecting the lower priority task space goals onto the null space of higher priorities:

$$\dot{\mathbf{q}}_{\mathbf{ref}} = \mathbf{S}(\mathbf{J}_{\mathbf{H}}^{\#}\dot{\mathbf{x}}_{\mathbf{H}} + \mathbf{N}_{\mathbf{H}}\mathbf{J}_{\mathbf{L}}^{\#}\dot{\mathbf{x}}_{\mathbf{L}}) \tag{5.2}$$

Where, $\mathbf{S} = \begin{bmatrix} \mathbf{I}_{n \times n} & \mathbf{0}_{n \times 6} \end{bmatrix}$ is the actuator selection matrix for (3.3), $\mathbf{J}^{\#}$ is the psuedoinverse of the Jacobian $\mathbf{J}$, $\dot{\mathbf{q}}_{\mathbf{ref}}$ is the reference joint velocity, and $\dot{\mathbf{x}}_{\mathbf{H}}$ and $\dot{\mathbf{x}}_{\mathbf{L}}$ are the high and low priority task space velocities, respectively. $\mathbf{J_H}$ are $\mathbf{J_L}$ are the corresponding high and low priority Jacobians, and $\mathbf{N_H} = \mathbf{I} - \mathbf{J_H^{\#}J_H}$ is the null space projection matrix. The reference joint velocities are integrated to obtain the reference command signal to be tracked by high gain local Proportional-Derivative (PD) controllers.

5.1.4 State Dependent Controllers

This section presents the specific controller formulation used during each state of the gait cycle.

STAND

The goal during this state is to maintain the COM position at the geometric centroid of both feet. In order to remain stable under small disturbances, the Jacobian under double support phase is used to compensate for the error $\Delta\mathbf{x_{COM}}$ in the x and y directions.

$$\begin{aligned} \mathbf{J_H} &= \begin{bmatrix} \mathbf{J_{STAND}} \\ \mathbf{J_{SWING}} \\ \mathbf{J_{COM}} \end{bmatrix} & \dot{\mathbf{x}}_\mathbf{H} &= \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \dot{\mathbf{x}}_\mathbf{COM} \end{bmatrix} \\ \mathbf{J_L} &= \begin{bmatrix} \mathbf{0} \end{bmatrix} & \dot{\mathbf{x}}_\mathbf{L} &= \begin{bmatrix} \mathbf{0} \end{bmatrix} \end{aligned} \tag{5.3}$$

PUSH

The goal during this state is to track the trajectory generated for $\mathbf{x_{COM}}$ to move to the stance foot support region while remaining in the double support phase. An augmented Jacobian matrix is used to track the trajectory while simultaneously maintaining the foothold constraints.

$$\begin{aligned} \mathbf{J_H} &= \begin{bmatrix} \mathbf{J_{STAND}} \\ \mathbf{J_{SWING}} \\ \mathbf{J_{COM}} \end{bmatrix} & \dot{\mathbf{x}}_\mathbf{H} &= \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \dot{\mathbf{x}}_\mathbf{COM} \end{bmatrix} \\ \mathbf{J_L} &= \begin{bmatrix} \mathbf{0} \end{bmatrix} & \dot{\mathbf{x}}_\mathbf{L} &= \begin{bmatrix} \mathbf{0} \end{bmatrix} \end{aligned} \tag{5.4}$$

The joint level reference velocities are calculated from (5.2) and integrated to obtain the position command.

LIFT

In the lift stage, the highest priority task is maintaining the foothold of the stance foot, holding the $\mathbf{x_{COM}}$ directly above it and simultaneously raising the swing foot from the ground. The key challenge in this state is that lifting the swing foot can potentially cause the centre of pressure to leave the support region formed by the contact points of the stance

foot. The prioritized task space control scheme is used to generate joint level commands to track the $\mathbf{x_{SWING}}$ trajectory while satisfying the higher priority goal of maintaining the foothold and balance.

$$\begin{aligned} \mathbf{J_H} &= \begin{bmatrix} \mathbf{J_{STAND}} \\ \mathbf{J_{COM}} \end{bmatrix} \dot{\mathbf{x}}_\mathbf{H} = \begin{bmatrix} \mathbf{0} \\ \dot{\mathbf{x}}_\mathbf{COM} \end{bmatrix} \\ \mathbf{J_L} &= \begin{bmatrix} \mathbf{J_{SWING}} \end{bmatrix} \quad \dot{\mathbf{x}}_\mathbf{L} = \begin{bmatrix} \dot{\mathbf{x}}_\mathbf{SWING} \end{bmatrix} \end{aligned} \tag{5.5}$$

The joint level reference velocities are calculated from (5.2) and integrated to obtain the position command.

SWING

At this point, the goal of the control approach is to generate a forward moving momentum along the selected sagittal plane. This deliberately destabilizes the biped by pushing $\mathbf{x_{COM}}$ outside the region of support in the chosen direction of motion. The task space prioritization in this state remains consistent with the previous state until the biped is unstable, at which point the control strategy enters the terminal **DROP** state.

$$\begin{aligned} \mathbf{J_H} &= \begin{bmatrix} \mathbf{J_{STAND}} \\ \mathbf{J_{SWING}} \end{bmatrix} \dot{\mathbf{x}}_\mathbf{H} = \begin{bmatrix} \mathbf{0} \\ \dot{\mathbf{x}}_\mathbf{SWING} \end{bmatrix} \\ \mathbf{J_L} &= \begin{bmatrix} \mathbf{J_{COM}} \end{bmatrix} \quad \dot{\mathbf{x}}_\mathbf{L} = \begin{bmatrix} \dot{\mathbf{x}}_\mathbf{COM} \end{bmatrix} \end{aligned} \tag{5.6}$$

The joint level reference velocities are calculated from (5.2) and integrated to obtain the position command.

DROP

In this terminal state, the Jacobian is used to track the fixed stance foot position and the generated swing foot trajectory to track the FPE point on the ground. Since the ZMP is outside of the region of foot support during this state, the torso is treated as a fixed base link and compute the Jacobian matrix of each foot.

$$\mathbf{J_H} = \begin{bmatrix} \mathbf{J_{STAND}} \\ \mathbf{J_{SWING}} \end{bmatrix} \dot{\mathbf{x}}_\mathbf{H} = \begin{bmatrix} \dot{\mathbf{x}}_{\mathbf{STAND}} \\ \dot{\mathbf{x}}_{\mathbf{SWING}} \end{bmatrix}$$
$$\mathbf{J_L} = \begin{bmatrix} \mathbf{0} \end{bmatrix} \qquad \dot{\mathbf{x}}_\mathbf{L} = \begin{bmatrix} \mathbf{0} \end{bmatrix} \tag{5.7}$$

CONTACT STABILIZATION

With an arbitrary 3D biped with finite sized (non-point) feet, it is possible for the biped to land on the edge of the foot instead of landing perfectly above the FPE point on the ground. Once ground contact is made, the solution to the FPE equation is no longer valid (since a real biped will not have instantaneous transfer of balance). To handle this behaviour, a stabilization substate is used where the joint level control is computed directly. At this point, trajectories are generated for the ankles to align the surface of the foot with the ground and switch to high gain PD control for tracking. This ensures that both feet are in full contact with the ground prior to executing the opposite leg's gait sequence. The biped-ground contact interface is shown before and after the stabilization substate in Figure 5.6.

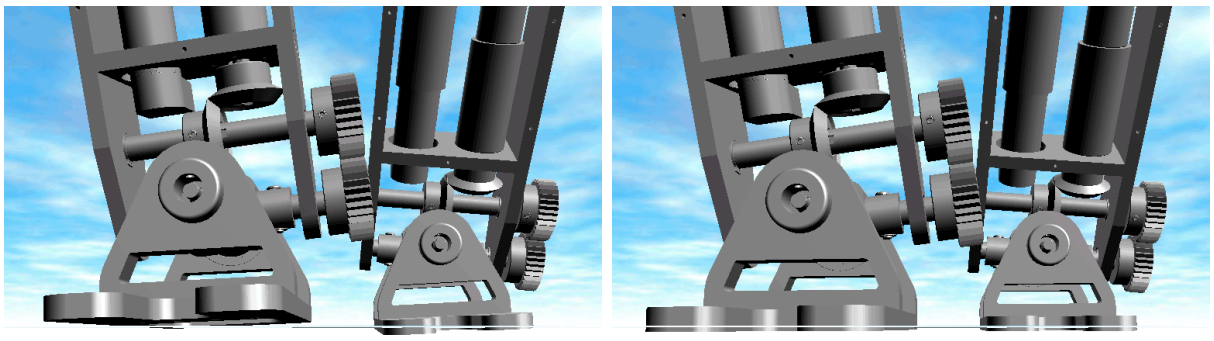


Figure 5.6: Ground-foot contact shown for before (left) and after (right) contact stabilization.

5.1.5 Computing the FPE Parameters

The 2D FPE equation (2.11) requires the total inertia about the COM (I_{COM}) and average angular velocity about the pivoted fixed foot ($\dot{\theta}_{avg}$). In the 2D case, the moment of inertia for link k is a scalar value since there is only one plane of rotation. The total inertia about the COM is computed by summing the moment of inertia for each link in the system. In the 3D case, the moment of inertia for each link is a 3×3 tensor:

$$\mathbf{I_k} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{5.8}$$

The inertia tensor of each link is taken at the COM aligned with the local coordinate system. If the xz-plane is selected as the sagittal plane in 3D space, the moment of inertia of link k is the I_{yy} term. However, in the 3D case the motion of the biped is no longer fixed to a single plane of rotation. By attaching a fixed coordinate frame to the selected sagittal plane at the start of each step, the orientation of the sagittal plane can be expressed as a 3×3 rotation matrix $\mathbf{R}$. The local inertia tensor can be rotated into the selected sagittal plane's coordinate frame by:

$$\mathbf{I_{k,sagittal}} = \mathbf{R} \cdot \mathbf{I_k} \cdot \mathbf{R^T} \tag{5.9}$$

Then the effective moment of inertia of each link *projected on to the selected sagittal plane* can be obtained by transforming the inertia tensor with (5.9) and pulling out the I_{yy} term, $I_{k,yy}$. The total inertia about the COM can then be computed by summing the effective inertia of each link on the sagittal plane:

$$I_{COM} = \sum_{k=1}^{n} I_{k,yy} \tag{5.10}$$

The average angular velocity is computed as a weighted sum of the inertia of each link [8]. In the 3D case, the same equation is used with the *effective* moment of inertia of each link on the sagittal plane:

$$\dot{\theta}_{avg} = \frac{\sum_{k=1}^{n} I_{k,yy}\dot{\theta}_k}{\sum_{k=1}^{n} I_{k,yy}} \tag{5.11}$$

The angular velocity of each link $\dot{\theta}_k$ is obtained by rotating the joint velocity (expressed as a 3×1 vector with $\dot{q}_k$ in the row represented by its local axis of rotation) to the fixed frame of the sagittal plane.

The parameters computed with (5.10) and (5.11) are plugged in to the 2D FPE equation (2.11) and the solution (ϕ) is obtained with a non-linear equation solver.

5.2 Simulations and Results

The proposed control strategy to extend the FPE theory to 3D was implemented in simulation on a 14 DOF lower body bipedal robot. Each state of the control strategy was implemented in the Matlab/Simulink environment with the multibody dynamics simulation by SimMechanics. Accurate kinematic and dynamic properties of the physical robot were taken directly from the CAD model through the toolchain model generation process (as discussed in Section 4.3). The controller diagram shown in Figure 5.7 implements the control strategy presented in this section. The subsystems from the outer most loop working inwards are described as follows:

1. **State Machine**
   Finite state machine logic was implemented in StateFlow as shown in Figure 5.8. The state names follow the same convention as the 2D FPE case (shown in Figure 2.9) while the transition logic is updated for the 3D case described in Section 5.1.4.

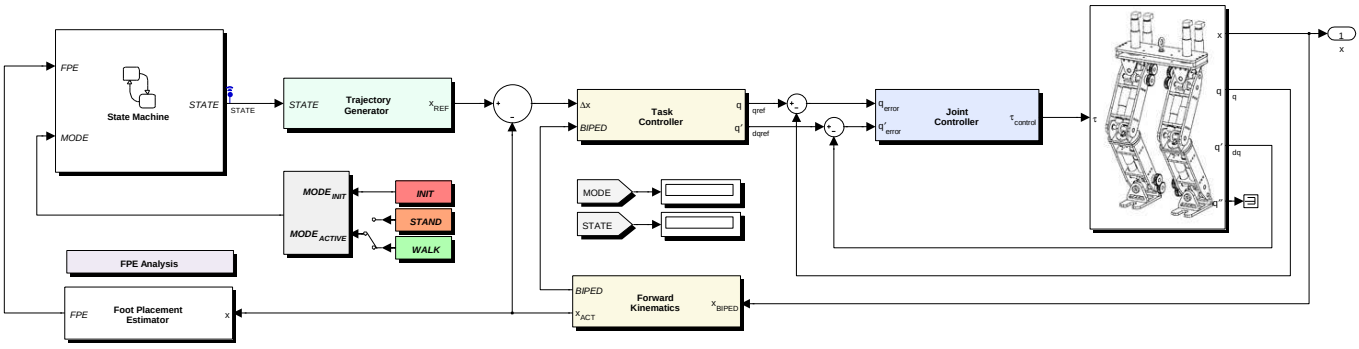


Figure 5.7: Controller diagram of 3D FPE based walking control strategy implemented in Simulink.

2. **Trajectory Generator**
   Generates the task space COM ($\mathbf{x_{COM}}$) and the swing foot ($\mathbf{x_{SWING}}$) trajectories detailed in Section 5.1.2 based on the current state of the controller. Also uses the higher level supervisory control mode (i.e. **WALK** or **STAND**) to adjust the trajectory generation for understepping/overstepping the FPE point.

3. **Foot Placement Estimator**
   Computes the FPE parameters about the COM (I_{COM}, $\dot{\theta}_{avg}$) for the 3D case (detailed in Section 5.1.5) and solves the FPE equation (2.11) using a numerical methods-based nonlinear solver from [7].

4. **Task Controller**
   Provides the state dependent control logic for the whole body motion control framework in Section 5.1.3. The resulting joint space velocities are integrated to obtain the joint position reference command. The contact stabilization substate used in the second half of **DROP** is also implemented here to generate joint level trajectories directly.

5. **Joint Controller**
   High gain joint level PD controllers with gain scheduling based on the current state of the control strategy. The output of this block ($\boldsymbol{\tau}$) is used to drive the forward dynamics simulation generated with the toolchain.

The simulations are executed with a fixed step Runge-Kutta solver at 1KHz. The forward dynamics generated with the toolchain drive the simultaneous 3D visualizations in an external viewer application.

5.2.1 Contact Modeling

The initial spring-damper contact model discussed in Section 3.1.3 was replaced with a more complex version to accurately model the ground/foot dynamics. The Hunt and Crossley contact model [113, 114] generates a normal force using a non-linear spring-damper system defined by:

$$F_{normal} = bz^p\dot{z}^q + kz^n \tag{5.12}$$

Where z, $\dot{z}$ are the penetration depth position and velocity, respectively. The constants k, b are the spring-damper coefficients and n, p, q are tunable constants. Equation (5.12) provides the ground reaction force in the normal direction only. The tangential (frictional) contact forces were modelled by:

$$F_{tangential} = f\dot{x} \tag{5.13}$$

Where $\dot{x}$ is the tangential velocity of the contact point (in the x and y directions) and f is a tunable constant. The forces generated by (5.12) and (5.13) ensure that there are no discontinuities when ground contact is made. This contact model was chosen as a good trade-off between accuracy and computational efficiency, and does not model all aspects of physical contact. A separate dynamic simulation model was generated to hand-tune the contact model constants to emulate a stiff ground. The final (tuned) parameters used for dynamic simulations are provided in Table 5.1.

It was found that stiffening the ground contact by raising the model constants would produce singularities with fixed time step simulations at 1 KHz. Reducing the time step helped avoid singularities but the simulation runtime drastically increased.

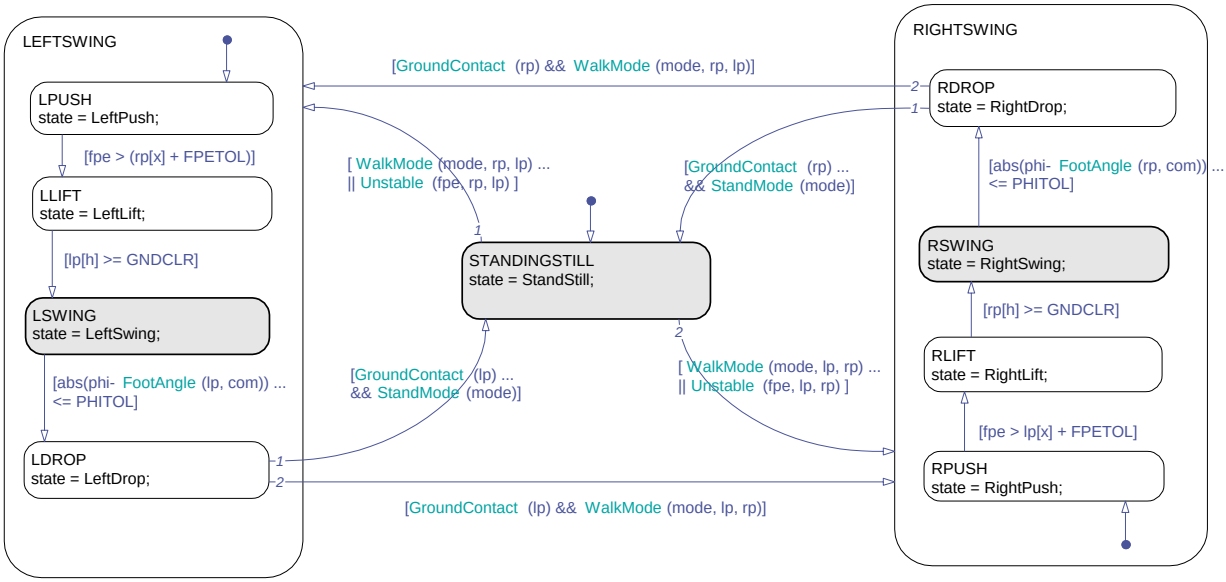


Figure 5.8: Finite state machine implemented in StateFlow for 3D FPE-based walking control strategy.

Table 5.1: Tuned contact model constants.

Parameter	Value
f	10
k	2000
b	10
p	1.10
q	1.00
n	2.31

5.2.2 Side-to-Side Stepping

To demonstrate the dynamic stability of a 3D biped under this approach, the frontal plane was selected as the plane for motion. Forward motion along this selected plane results in a side-to-side stepping sequence for the biped (as shown in Figure 5.9). The gray plane in the frame captures moves along the y-direction (biped's frontal plane). The intersection of the gray plane and the ground indicates the FPE point tracked during **DROP**.

The resulting $\mathbf{x_{COM}}$ trajectories from simulating the side-to-side stepping motion (shown in Figure 5.10) demonstrate the stability of the biped through a complete gait sequence. Note that in side-to-side stepping, the biped simply rocks back and forth (i.e. there is no forward progress along the plane of motion). The oscillations at 12s and 25s are due to the contact stabilization substate (as shown in Figure 5.6) attempting to evenly distribute the ground reaction force over soles of both feet against a soft contact model.

The prioritized motion control framework handles the dynamic switching of constraints (from double support to single support) while generating the appropriate joint level commands for swinging the COM over. The colour coded dotted lines on Figure 5.10 indicate the boundaries of each foot on the ground. Note that during the **SWING** state, the COM is pushed outside the region of support (around 11s). This in turn initiates the **DROP** state where the FPE point is tracked to regain stability.

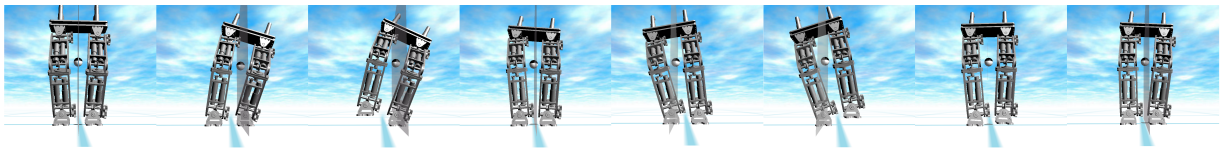


Figure 5.9: Frame captures from the real-time 3D visualization while side-to-side stepping.

In the terminal **DROP** state, the swing foot trajectory tracks the FPE point on the ground (shown in Figure 5.11) with an added offset to ensure that the biped oversteps to guarantee stability (as per the 2D FPE theory). Once ground contact is made (around 11.6s), the stabilization substate is entered and the swing foot trajectory is controlled directly to align the foot with the ground. This causes the biped to rock back and forth (similar to the 2D case) until stability is reached.

5.2.3 Forward Walking Gait

Forward walking is achieved by selecting the biped's sagittal plane as the direction of motion. The frame captures from forward walking simulations are shown in Figure 5.12. The key challenge is maintaining stability in the off-sagittal plane while the swing foot is raised off the ground during the **LIFT** state. The $\mathbf{x_{COM}}$ trajectories shown in Figure 5.13

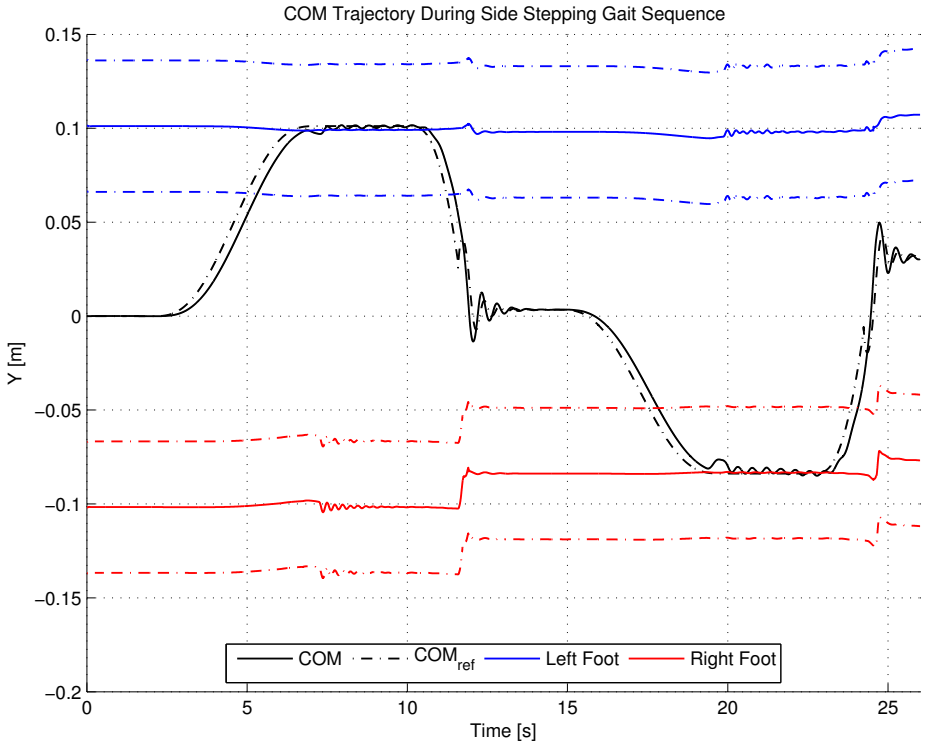


Figure 5.10: COM trajectory being tracked during the complete gait sequence of side stepping.

demonstrate the dynamic stability in the off-sagittal plane while forward progress is made along the chosen direction of motion. The gait cycle in the forward walking direction is slow due to the use of a soft contact model to enable simulations at 1 KHz. The softer contact model also requires a longer stabilization period (around 5s) once ground contact is made to ensure that the foot is perfectly stable and aligned with the ground plane before proceeding. The proposed trajectory generation for $\mathbf{x_{COM}}$ illustrated by Figure 5.4 can be seen at 8s and 22s when the COM is pushed outside the region of support once the swing foot is aligned.

The alternating foot trajectories on the selected sagittal plane during a complete gait cycle are shown in Figure 5.14. The first step is taken with the left foot tracking $\mathbf{x_{SWING}}$ (between 5-12s). The **DROP** state is entered around 12s and the left foot tracks the $\mathbf{x_{SWING}}$ trajectory given by $\mathbf{FPE} + \mathbf{FPE_{offset}}$ to ensure over stepping. Once ground contact is made, the contact stabilization substate is used to straighten both feet and

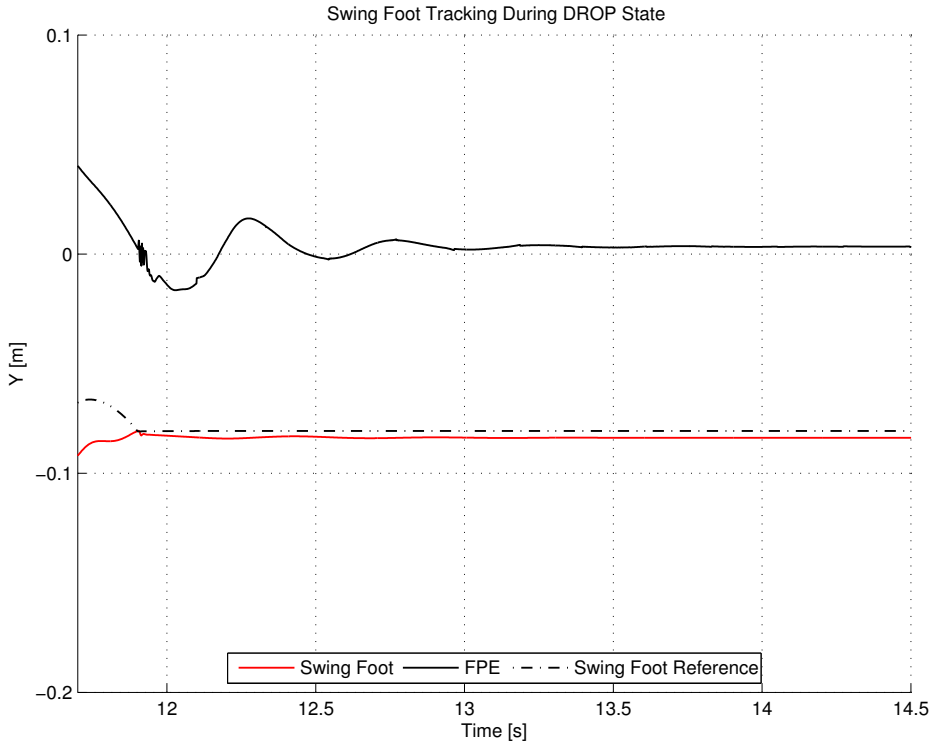


Figure 5.11: Swing foot tracks a point on the ground given by $\mathbf{FPE} + \mathbf{FPE_{offset}}$ to ensure overstepping.

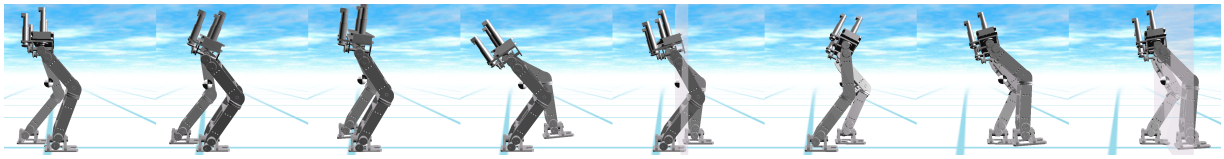


Figure 5.12: Frame captures from the real-time 3D visualization with forward walking gait.

spread the ground reaction force evenly (around 12-14s). The step cycle begins for the right leg from 15s to complete the gait sequence.

The FPE trajectory along the forward walking direction is shown in Figure 5.15. The FPE location on the ground is tracked (with an offset) to regain stability around 12s by the swinging left foot and again around 25s by the swinging right foot. The FPE location also

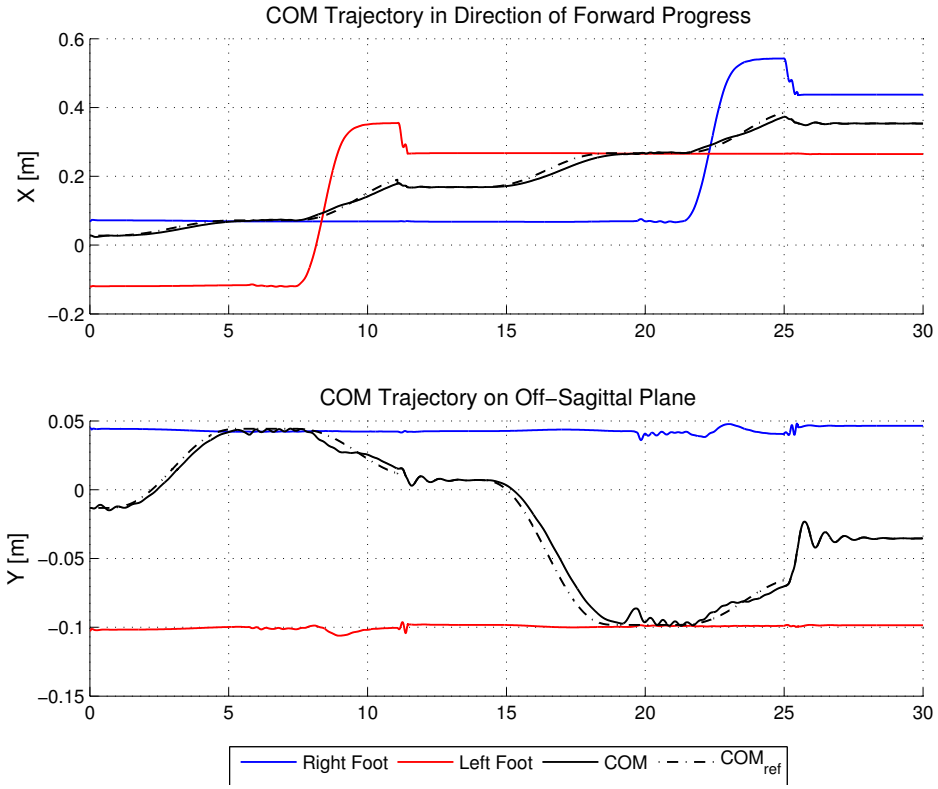


Figure 5.13: COM and foot trajectories in the direction of forward progress and off-sagittal plane during a complete gait cycle.

remains safely between the feet during the double support phase after each step is made (approximately 13-15s).

5.3 Summary

The FPE theory is extended to the 3D case to form dynamically stable gait cycles by selecting a 2D sagittal plane parallel to the desired direction of motion. The primary goal in the 3D case is to generate a forward momentum along this selected sagittal plane. For the duration of the step cycle, the FPE equation is used to determine the appropriate swing foot placement to restore balance from the momentum generated along the plane. A new sagittal plane is selected with each step to enable turning.

Trajectories are generated for the swing foot and COM position to simultaneously generate the forward momentum along the plane while remaining stable in the off sagittal plane until the **DROP** state is reached. Upon entering this terminal state, the 3D biped is falling forward in the desired direction of motion and the swing foot tracks the FPE point to restore balance.

A whole body motion control framework is used to generate the appropriate joint level trajectories during each state of the proposed control strategy. The framework uses a

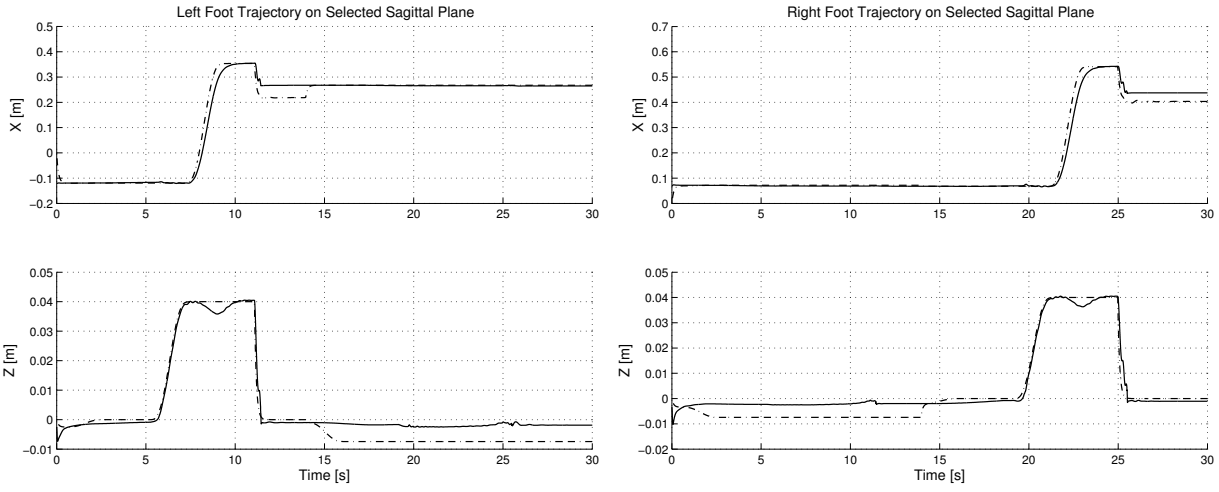


Figure 5.14: Left and right foot trajectories on the selected sagittal plane during a complete gait cycle.

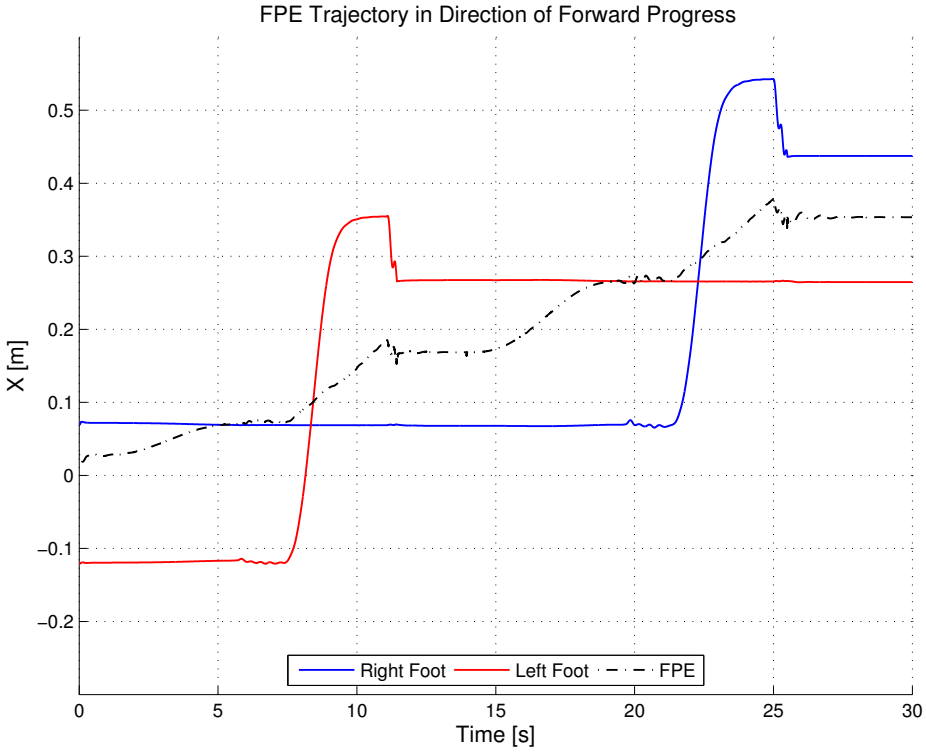


Figure 5.15: FPE and foot trajectories in the direction of forward progress during a complete gait cycle.

prioritized control scheme where the low priority constraints of each state are projected into the null space of higher priority tasks. This approach handles the dynamic switching of constraints as the biped moves between the single and double support phases.

The 14 DOF bipedal robot developed in previous chapters is used to demonstrate the control strategy presented in this chapter. Dynamic simulations with 3D visualization are generated directly from the CAD model using the toolchain. The simulations are used to verify the efficacy of the proposed control strategy for side-stepping and forward walking gait. Despite using a more complex contact model, tuning the parameters to accurately model a stiff ground in reality proved to be challenging. This imperfection in the simulation environment highlights the need to develop physical hardware to validate walking control strategies, as presented in the following chapter.

Chapter 6

Experimental Results

This chapter presents the experimental work completed to validate the actuator dynamics and motion control framework. The hardware platform used for experimental validation is presented in Section 6.1. This includes the mechanical implementation of the bipedal robot design and the control electronics used to control the robot.

The simulations used to demonstrate the proposed 3D FPE walking control strategy in the previous section accounted for link-side dynamics only. Section 6.2 describes the modifications to account for the actuator dynamics. This modification to the simulations enables a single controller designed in Simulink to target either the simulation environment or the physical hardware using the HIL architecture discussed in Section 6.3. The experimental validation for a single actuator and the proposed motion control framework are presented in Sections 6.4 and 6.5, respectively.

6.1 Physical Hardware

The electromechanical design presented in Chapter 3 was realized to develop the 14 DOF bipedal robot. The assembled 7 DOF leg of the biped is shown in Figure 6.1. The mechanical implementation was derived directly from the revised CAD models. The control electronics were developed by an industry partner, Quanser Inc. This section presents the realization of the physical hardware platform used for experimental validation.

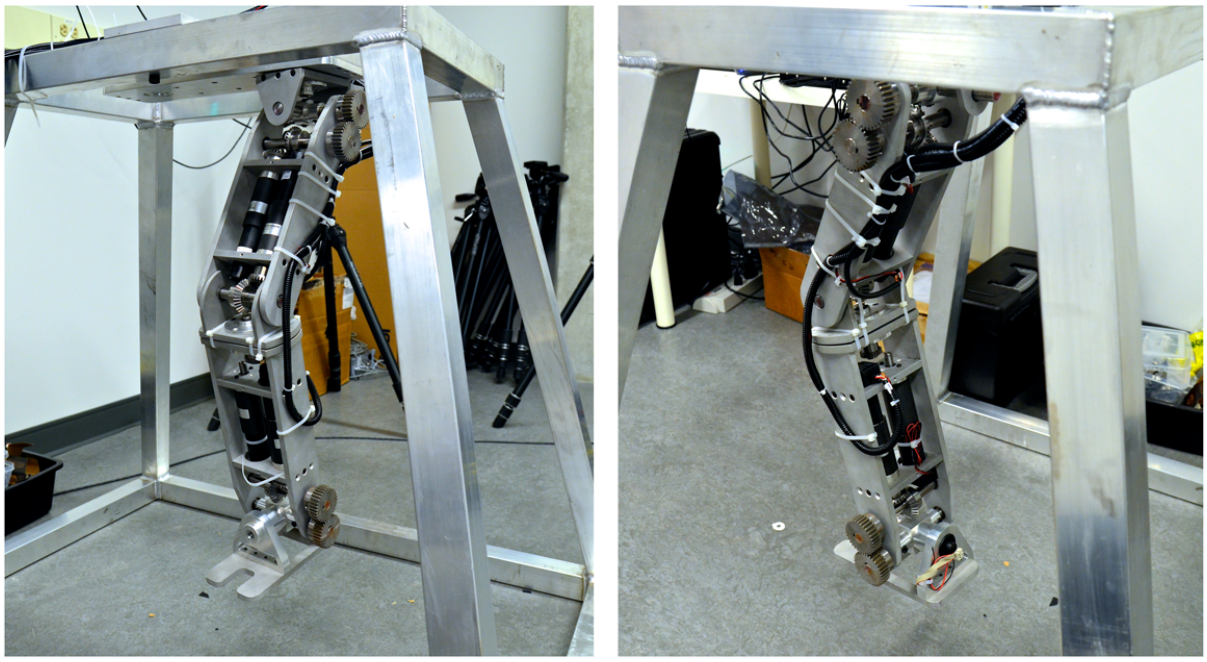


Figure 6.1: The 7 DOF leg built for the bipedal robot based on the electromechanical design in Chapter 3.

6.1.1 Mechanical Implementation

The final mechanical design of the bipedal robot was developed on campus at the University of Waterloo. The on-site engineering machine shop manufactured the mechanical chassis components from the final CAD drawings (shown in Appendix A.1). The selected DC motors and gearhead combinations were sourced from a motor manufacturer. The drivetrain components (e.g. bearings, shafts, gears) used to relocate the output gearhead shaft from each joint axis were also sourced from a hardware vendor. The author of this thesis completed the mechanical assembly using the CAD model as a reference. The assembled joints and links of the bipedal robot leg are shown in more detail in Figure 6.2.

The mechanical chassis was composed primarily out of Aluminum 5052 and created using Computer Numerical-Controlled (CNC) machine tools directly from the CAD files. This enabled repeatable and accurate positioning of mechanical features in the chassis components (i.e. for the separation of holes for gearing components to mesh well). CNC machine tools also facilitated the production since some chassis components (e.g. motor mounts)

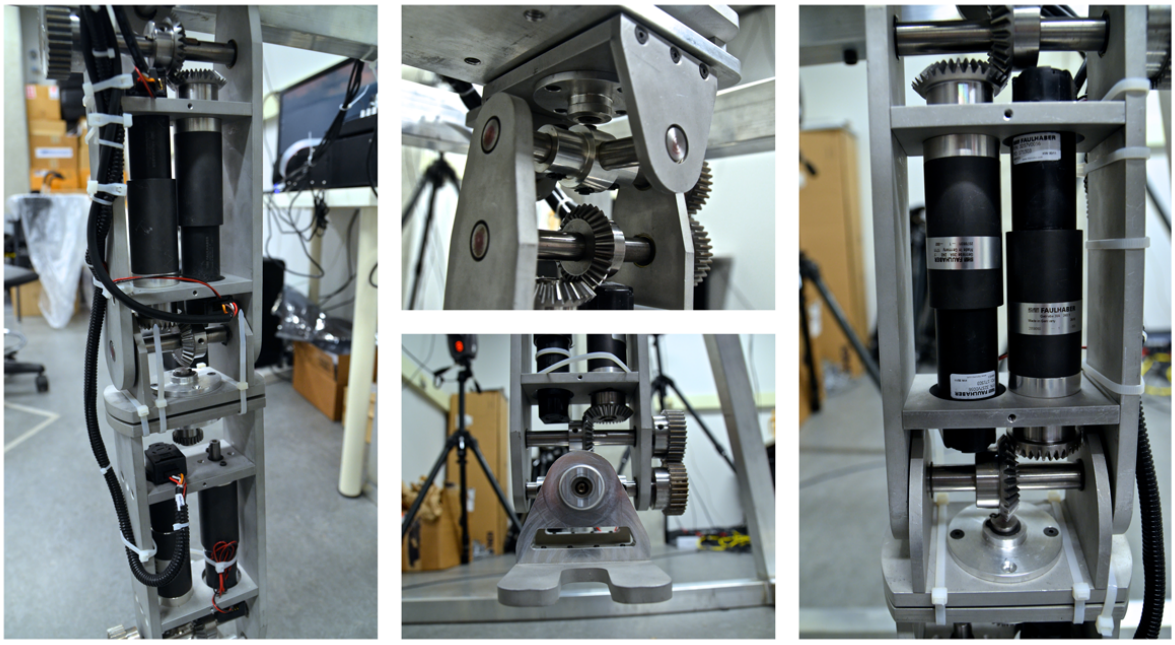


Figure 6.2: Close up pictures of various joints and linkages in the mechanical implementation of the bipedal robot design.

were used multiple times in the design of each leg. The advantage here was that once the machine was setup to produce the first component, the remaining copies could be produced faster using the same setup.

One of the key challenges in the assembled leg was the use of bevel gears for perpendicular mechanical coupling. It was difficult to position the gear axis with enough accuracy to achieve perfect clearance between mating components. As a result, some of the joints which use this design approach to shift the weight distribution suffer from backlash (or "joint play"). In the current design, this is particularly evident for the knee joint, where one of the stock gearing components had to be modified due to clearance issues.

In addition to developing the bipedal robot, a supporting frame was also designed for fixed and walking experiments (CAD drawings available in Appendix A.2). In the fixed configuration, the biped's torso is mounted directly to the frame. This provides sufficient ground clearance for the legs to remain suspended in the air. This removes the $n = 6$ unactuated DOF from the floating base but provides an experimental platform for testing basic control strategies (i.e. controlling swing foot motion). The walking configuration was

designed to provide sufficient clearance for the biped to be mobile, while using a tether to provide support in case of a fall.

6.1.2 Control Implementation

The electronics used for controlling the bipedal robot were developed by an industry partner. Quanser Inc developed custom motor controllers and drivers for the current requirements derived from torque estimates in dynamic simulations. Each joint in the bipedal robot has a dedicated local motor controller and driver unit. A control system model developed in Matlab/Simulink using Quanser's QUARC toolbox communicates with the electronics hardware through a serial interface (USB). Quanser's system enables hardware-

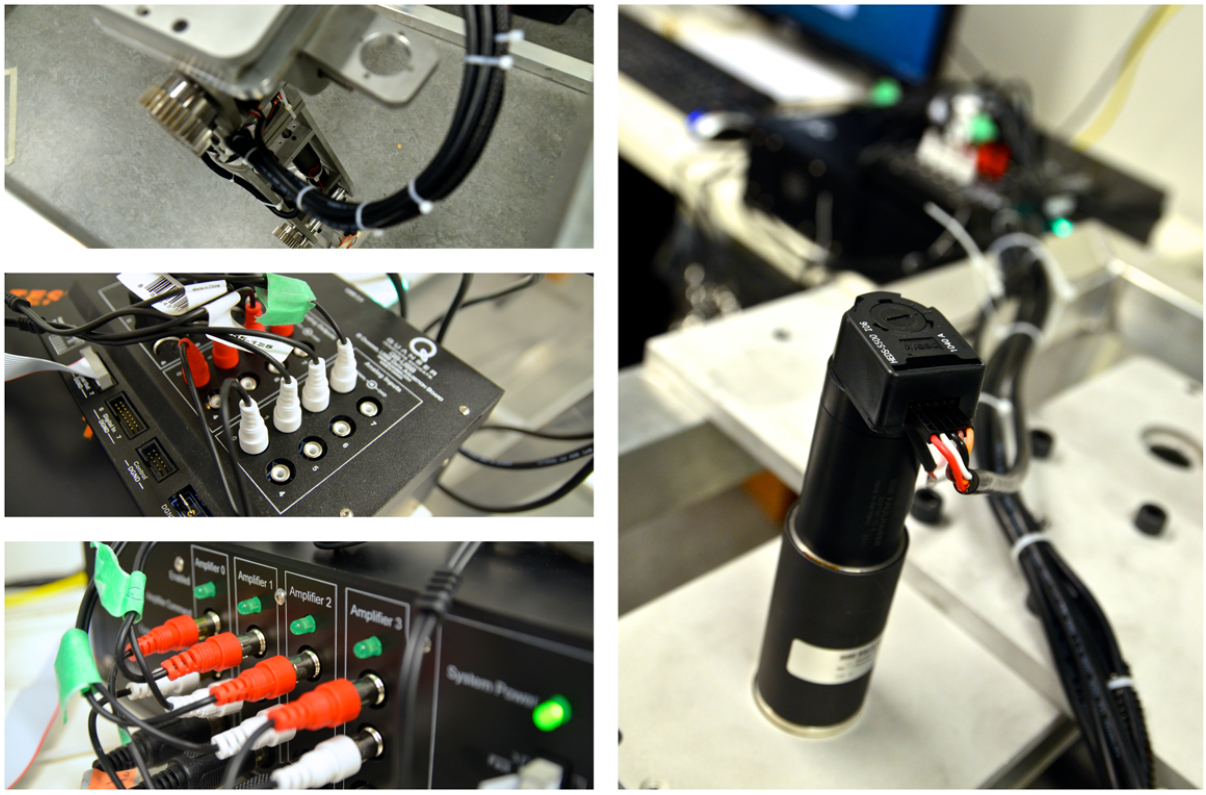


Figure 6.3: Electronics and wiring used to interface control loops running in Simulink to the physical hardware.

in-the-loop (HIL) experiments with the physical bipedal robot (described further in Section 6.3).

Unfortunately, the electronics hardware developed by Quanser was not ready in time for experimental validation. Since it is a custom solution, further development was required before it could be used to control the complete 14 DOF bipedal robot. Quanser provided the author with control electronics from their existing product offerings in the interim to complete *some* experimental validation on the physical robot. The general idea is that the HIL experiment models developed to target the interim solution could easily be switched to target the custom solution once it is ready. The interim solution is constrained to 7 channels. A single 7 DOF leg shown in Figure 6.3 was wired to these channels for the experimental validation in this chapter.

6.2 Actuator Model

PD control is applied for each joint in order to track the desired trajectory generated by higher levels of control (Section 5.1.3). The control signal u produced at each joint k is provided by (where $k = 1 \dots n$ is the k-th joint of the n-DOF system):

$$u_k = K_P(q_{d_k} - q_k) - K_D \dot{q}_k \tag{6.1}$$

Where q_{d_k} and q_k are the desired and actual angles and $\dot{q}_k$ is the velocity of the k-th joint. Constants K_P and K_D are the porportional and derivative gains of the controller, respectively. In the ideal case, the control signal u_k would simply be equal to the applied torque τ_k of each joint (which forms $\boldsymbol{\tau} = [\tau_1 \dots \tau_k]$ shown on the right hand side of (3.2)). However, the actuator dynamics of DC motors used in the development of the 14 DOF bipedal robot must be considered. The motors selected in Section 3.2.5 are controlled by a voltage control signal v_m. A second order system is used to model the actuator dynamics and relate the applied torque to the motor voltage v_m [115]:

$$J_m \ddot{\Theta}_m + \left(B_m + \frac{k_b k_m}{r_a} \right) \dot{\Theta}_m = \tau_m - \frac{\tau_l}{g_r} \tag{6.2}$$

Where Θ_m is the rotor angle, J_m is the motor inertia, B_m is the motor damping, k_b is the back emf or voltage constant, k_m is the torque constant, r_a is the armature resistance and g_r is the gear reduction ratio. The motor torque τ_m is related to the (link-side) load torque

τ_l through the gearing ratio g_r. Since the output side of the gearhead is coupled directly to the link, the motor angles are related to the joint angles by:

$$\Theta_{m_k} = g_{r_k} q_k \tag{6.3}$$

Similarly, the joint torques in $\boldsymbol{\tau}$ (3.2) are related to the actuator load torques by:

$$\tau_{m_k} = \frac{\tau_k}{g_{r_k}} \tag{6.4}$$

Furthermore, the motor torque is related to the applied voltage through the following relationship:

$$\tau_{m_k} = k_m i_a = \left(\frac{k_m}{r_a} \right) v_{m_k} \tag{6.5}$$

Where i_a is the current through the armature wiring. Substituting equations (6.3 - 6.5) back into (6.2) yields the complete relationship between the k-th joint angle, link-side torque and the applied motor voltage:

$$g_r^2 J_m \ddot{q}_k + g_r^2 \left(B_m + \frac{k_b k_m}{r_a} \right) \dot{q}_k = g_r \left(\frac{k_m}{r_a} \right) v_k - \tau_k \tag{6.6}$$

Note that the drivetrain constants specific to the k-th joint are used (e.g. J_{m_k}, B_{m_k}, g_{r_k}, etc.) in this equation (6.6).

6.2.1 Independent Joint Control

Using independent joint control [116], each joint k of the system is decoupled from the rest of the system and controlled individually. The control signal for each joint is computed directly from its own reference trajectory, position and velocity. This approach does not account for the coupled dynamics of the overall system described by (3.2). The link-side torques in (6.6) are treated as a disturbance to the second order system and the motor inertia and damping are modified as follows:

$$\begin{aligned} J_{eff} &= g_r^2 J_m + a_{kk} \\ B_{eff} &= g_r^2 (B_m + k_b k_m / r_a) \end{aligned} \tag{6.7}$$

Where J_{eff} and B_{eff} are the effective motor inertia and damping *seen* by the joint. The a_{kk} in (6.7) compensates for the inertia of link k by adding the k-th diagonal term from the inertia matrix $\mathbf{A}(\mathbf{q})$ in (3.2). Substituting back into (6.6) yields:

$$J_{eff}\ddot{q}_k + B_{eff}\dot{q}_k = g_r \left(\frac{k_m}{r_a} \right) v_k - d_k \tag{6.8}$$

Where $d_k = \tau_k / g_r$ is the link-side torque treated as a disturbance to the system. Taking the control input u_k (6.1) to be the voltage signal v_m in (6.8) yields a closed loop controller for independent joint control with actuator dynamics (implemented in Figure 6.4).

To improve the estimate of the motor side inertia, a_{kk} is required in (6.7). The dynamic simulation package used in Section 5.2 is based on the Matlab/Simulink environment and uses the SimMechanics toolbox which does not allow the mass matrix $\mathbf{A}(\mathbf{q})$ to be isolated. Algorithm 1 was used to compute the mass matrix and extract the diagonal terms at each time step.

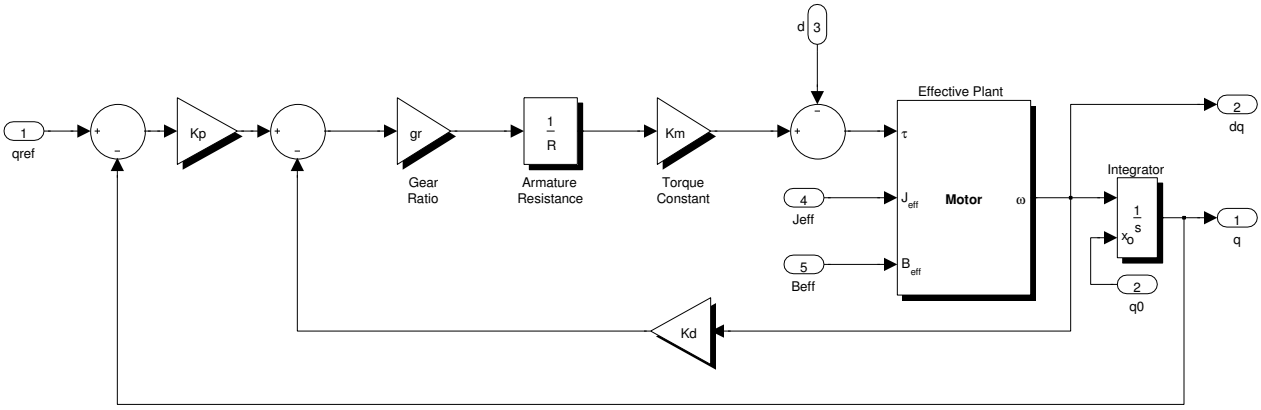


Figure 6.4: PD controller model for independent joint control with actuator dynamics.

Algorithm 1: Computing mass matrix diagonal terms with RNE algorithm

```
Initialization;
Set q = q_0, q̇ = q̇_0, q̈ = 0;
while 1 do
    Compute τ̂ using RNE with q̈ = 0;
    for i = 1 to n do
        Set g̃ = 0, q̃̇ = 0, q̃̈_i = 1, q̃̈_j = 0 [j ≠ i];
        Compute τ̃ using RNE;
        Form i-th column: A(q)_i = τ̃ − τ̂
    end
    Combine columns to form A(q);
    Select diagonal elements of A(q);
end
```

6.3 HIL Architecture

The architecture used to control the physical 14 DOF bipedal robot is presented in this section. In general, controlling DC motors requires a controller (to host the control algorithm) and a driver (typically serves as a power amplifier). The controller outputs a low voltage control signal which is amplified by the driver and applied to the terminals of the motor. Encoders are used to sense the rotor position used by the controller for closed loop control.

The physical bipedal robot in Section 6.1.1 is treated as the plant. Using Quanser's QUARC toolbox, control algorithms developed in the Simulink environment can be compiled for a variety of target platforms ranging from embedded microcontrollers to standard PC. This allows the use of a shared codebase for simulation and for the physical hardware. The control algorithms used for the biped are compiled and executed on a PC which communicates with the physical hardware through a data acquisition (DAQ) board. A separate and dedicated voltage amplifier is used to drive the DC motors from a low voltage analog control signal (labeled "amplifier command" on Figure 6.5) from the controller via the DAQ. As an interim solution, Quanser provided the author with a Q8-USB (DAQ board) and a VoltPAQ (voltage amplifier) shown in Figure 6.3.

The actuator subassemblies selected in Section 3.2.5 were preassembled with incremental optical encoders mounted to the motor side. These encoders output digital signals on two

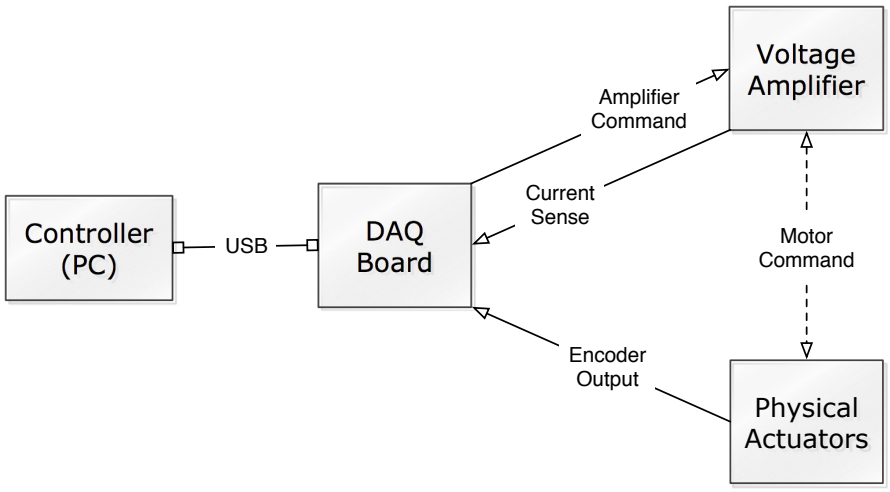


Figure 6.5: Hardware architecture used to control the physical bipedal robot leg.

channels. These digital signals are passed back to the controller running on the PC via the DAQ. Using quadrature decoding, the motor angle Θ_m can be retrieved from the encoder output in degrees:

$$\Theta_m = \frac{360}{n \cdot l} \cdot Encoder_{output} \tag{6.9}$$

Where $n = 4$ for quadrature (4X) decoding and l is the lines per revolution specified from the encoder manufacturer. Using the relationship between the motor variables and the joint variables in (6.3), an estimate of the k-th link-side angle (at the output of the gearhead) can be obtained:

$$q_k = \frac{360}{4 \cdot l \cdot g_{r_k}} \cdot Encoder_k \tag{6.10}$$

This is only an estimate of the joint angle since there are drivetrain losses (i.e. in the gearhead) which are ignored by an encoder mounted to the motor side.

6.3.1 Parallel Models

The ability to develop a single controller and target either a simulation environment or the physical hardware is extremely useful. Using the Quanser's QUARC toolbox, a Q8-USB

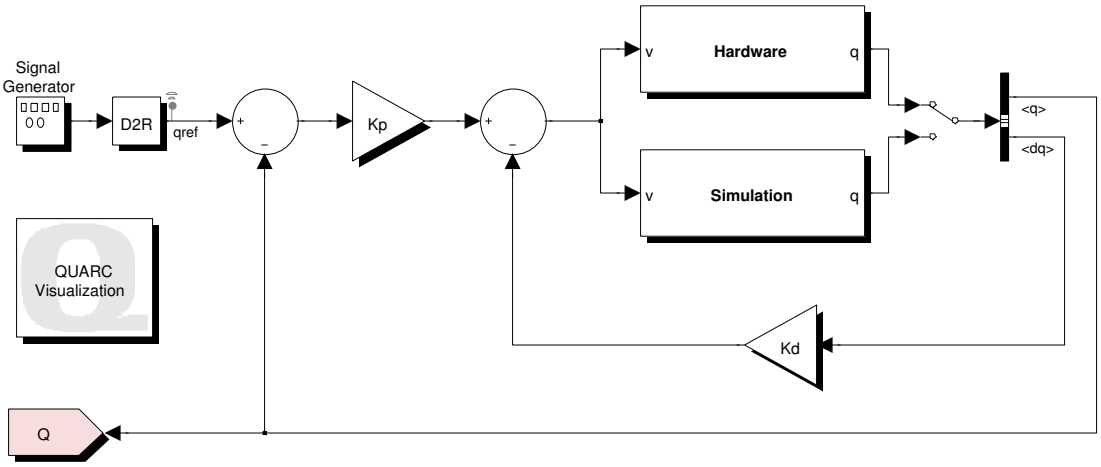


Figure 6.6: Parallel models designed to target either simulations or physical hardware with the same controller.

(DAQ) board and a VoltPAQ (voltage amplifier), the control algorithms were redeveloped as parallel models capable of switching target platforms (shown in Figure 6.6).

In the physical environment model shown in Figure 6.7, the hardware target subsystem is initialized to interface with the Q8-USB DAQ board. The control algorithm running on the PC communicates with the physical hardware using "HIL Read" and "HIL Write" blocks which communicate with the DAQ over USB. The PD controller output is an analog signal corresponding to the desired voltage at the motor terminals (amplifier command). The DAQ also reads the raw digital signals from the motor side encoders. The VoltPAQ voltage amplifier also provides the ability to read the current in the armature circuit (i_a) of the DC motor. This value is passed back to the controller through the DAQ using an analog channel. Due to the noise in the current sensor, a low pass filter is needed. The filter parameters were provided by Quanser. The quadrature decoding formula (6.10) is used to obtain the joint angle for the closed loop feedback. The link-side angle information is also used to determine whether a joint is out of its limits. The controller detects and sends a "shut down" command to the amplifier channel if it is.

In the simulated environment mode, the controllers were reformulated to use the voltage signal as the control input. The simulation target subsystem shown in Figure 6.8 models the actuator dynamics and treats the link-side torques as a disturbance input. The VoltPAQ amplifier gain (3 V/V) is applied to the voltage control signal to model the effective voltage at the motor terminals. The actuator dynamics are simulated with the motor (r_a, k_m, k_b,

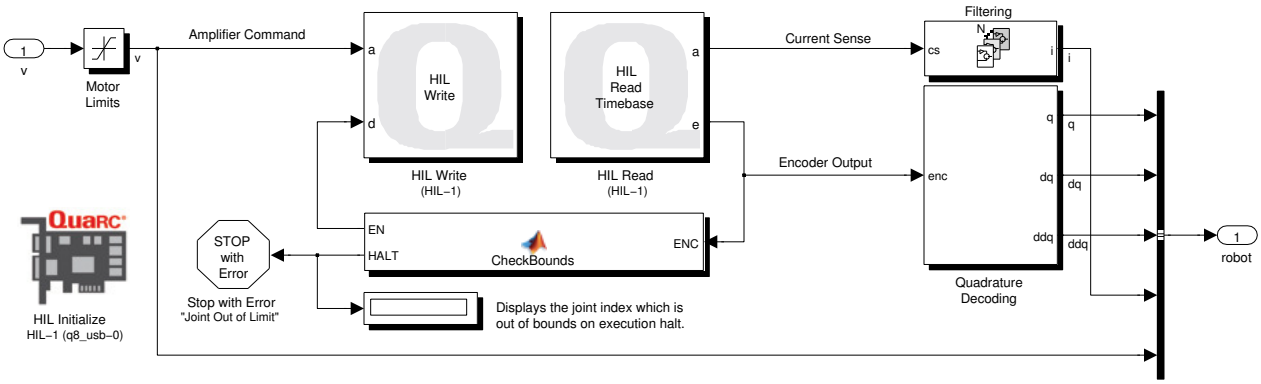


Figure 6.7: HIL subsystem from Figure 6.6 used to target physical hardware with voltage control signal.

J_m and B_m) and gearhead (g_r) parameters. The link-side variables are obtained by applying the gear ratio to the motor velocity $\dot{\Theta}_m$ to obtain $\dot{q}$. The joint positions and accelerations are computed by taking the integral and derivative of $\dot{q}$, respectively. The joint angles, velocities and accelerations (q, $\dot{q}$, $\ddot{q}$) are used with inverse dynamics to compute the effective link-side torques, which is fed back as a disturbance input to the simulated motor model.

Note that the full dynamic model of the 7 DOF leg is used for the plant in the simulation part of the experiments. The simplified dynamics are only used by the controller.

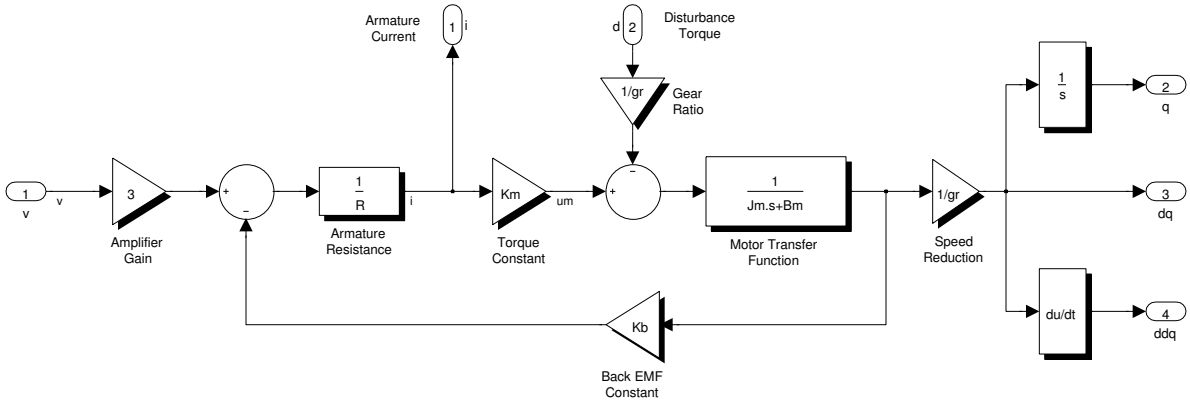


Figure 6.8: Subsystem from Figure 6.6 used to target the simulated environment with voltage control signal.

6.4 Single DOF Validation

This section describes the experiments validating the simulation models for single DOF controllers using the HIL environment. Parallel models developed in Section 6.3.1 were used to compare the results from simulation to the physical hardware.

6.4.1 Joint Tracking

Local PD controllers are used to track the joint level trajectories. The parallel model shown in Figure 6.6 was developed to tune the proportional (K_P) and derivative (K_D) gains for individual joints on the bipedal robot. First, the gains were tuned in the simulation environment to achieve the desired tracking performance. The shared controller architecture was then switched to target the physical hardware to validate the tracking results.

The experimental results presented in this section compare the tracking performance of three joints in simulation and in hardware. Each joint tracks a sinusoidal reference trajectory at $0.2Hz$. The amplitude and bias of each reference trajectory is selected to maintain the motion within the joint limits. Figures 6.9, 6.10 and 6.11 compare the tracking performance for the hip yaw, hip pitch and knee pitch joints, respectively.

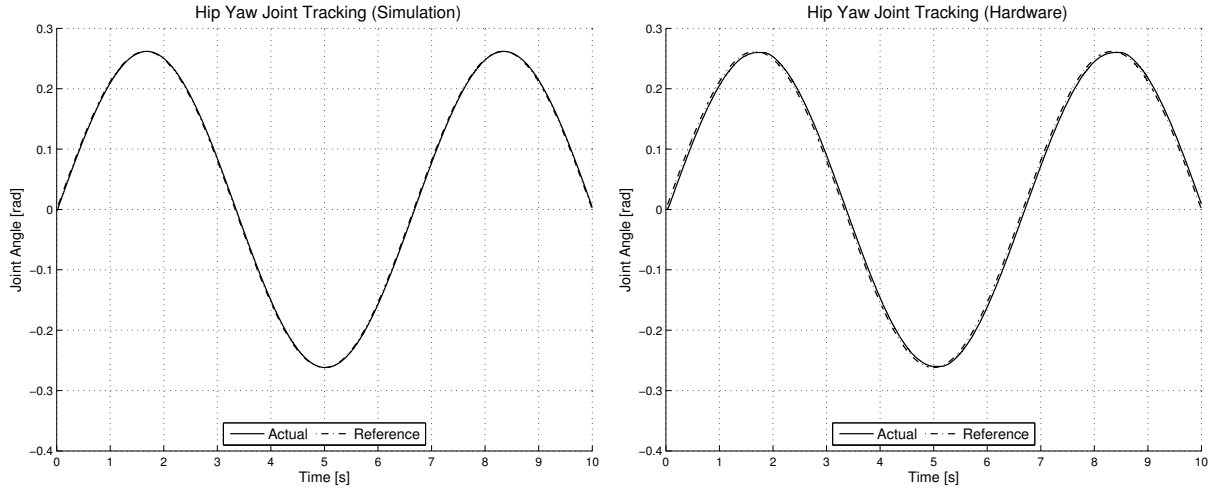


Figure 6.9: Hip yaw joint tracking results for simulation and hardware.

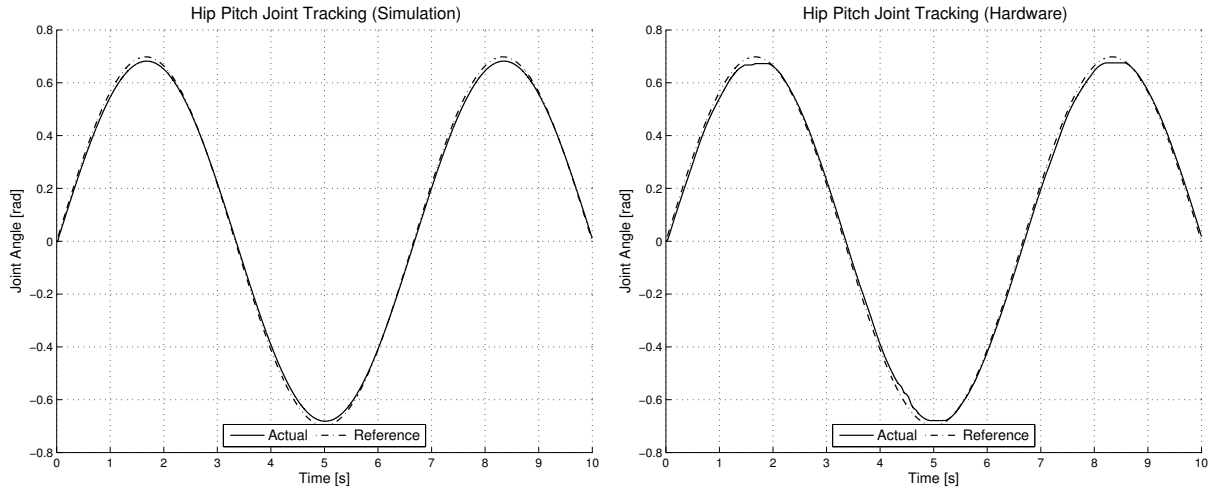


Figure 6.10: Hip pitch joint tracking results for simulation and hardware.

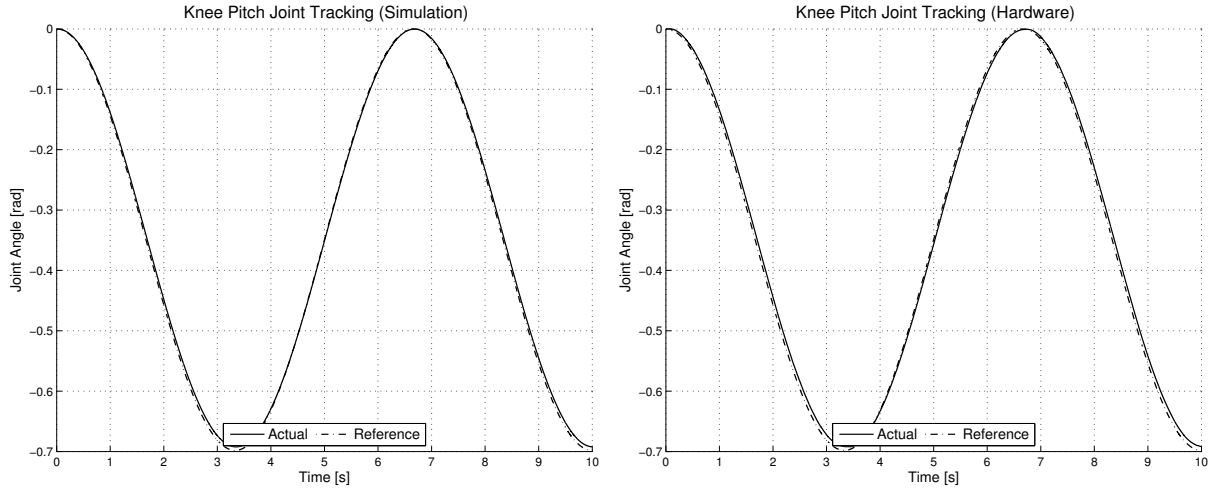


Figure 6.11: Knee pitch joint tracking results for simulation and hardware.

With the tuned PD gains, the tracking performance on the physical hardware is nearly identical to the simulation environment. Hip yaw joint tracking (Figure 6.9) shows that there is a very small but negligible lag on the physical hardware when compared to the simulated system. Hip pitch joint tracking (Figure 6.10) shows perfect tracking for the

simulated system while there are small disturbances near the peaks of the sinusoidal wave on the physical hardware. This discrepancy is due to the backlash problem discussed in Section 6.1.1. The simulated system demonstrates perfect tracking since backlash is not modeled. The backlash in the joints further down the kinematic chain affects the tracking performance of the proximal joints. For example, the play in the knee joint introduces a varying torque loading on the hip joint as it moves through the full range of motion. However, the PD controller gains are sufficiently high for the physical system to overcome the disturbances due to backlash and achieve near perfect tracking.

6.5 Motion Control Validation

This section describes the experiments validating the motion control framework in Section 5.1.3 using the HIL environment. The parallel model concept was extended to develop higher level task space controllers to generate the joint level trajectories for simulation and hardware. The interim control electronics provided by Quanser Inc are limited to controlling 7 DOF simultaneously. Therefore, only a single 7 DOF bipedal robot leg wired to the control electronics was used for the experiments in this section. The biped's torso was rigidly attached to the supporting frame in the fixed configuration, removing the 6 unactuated DOF. The Jacobian matrix used to map the task space and joint space velocities in (5.1) is reduced to:

$$\mathbf{J} = \left[\partial \mathbf{q_{act}}\right]_{m \times n} \tag{6.11}$$

$$\dot{\mathbf{q}}_{\mathbf{ref}} = \mathbf{J}^{\#} \dot{\mathbf{x}}_{\mathbf{ref}} \tag{6.12}$$

Where $n = 7$ is the actuated and controllable DOF supported by the control electronics. The joint level commands are generated from the task space trajectories ($\dot{\mathbf{x}}_{\mathbf{ref}}$) using (6.12). Alternatively, the prioritized task space control scheme in Section 5.1.3 can be used to simultaneously satisfy high priority constraints. The mapping provided by (5.2) is simplified by removing the actuator selection matrix $\mathbf{S}$, since there are no unactuated DOF.

The fixed configuration of the supporting frame and high gearing ratio of each joint causes the 7 DOF leg to naturally hang straight at the start of each experiment (i.e. $\mathbf{q} = \mathbf{0_{7\times 1}}$). This represents a singular configuration of the leg. In Section 6.4, the joint level commands were provided directly to the local PD controller. Computing the joint level commands with (6.12) requires inverting the Jacobian matrix, which is problematic for singular configurations. To address this issue, each experiment starts by directly controlling the joint level trajectories for $t_{init} = 5$ seconds to manually move the leg into a non-singular configuration.

6.5.1 Planar Motion Control

Some motions during the bipedal gait cycle can be simplified to a planar motion control problem. Common motions during the swing phase include bending of the knee joint and tracking an arc-like trajectory for the swing foot position. For these cases, the 3 pitch joints in the 7 DOF leg are primarily used to carry out the desired motion. A simplified version of the motion control framework (6.12) can be used to control the swing foot position in the biped's sagittal plane.

$$\dot{\mathbf{q}}_{\mathbf{ref}} = \mathbf{J}^{\#}\dot{\mathbf{x}}_{\mathbf{SWING}} \tag{6.13}$$

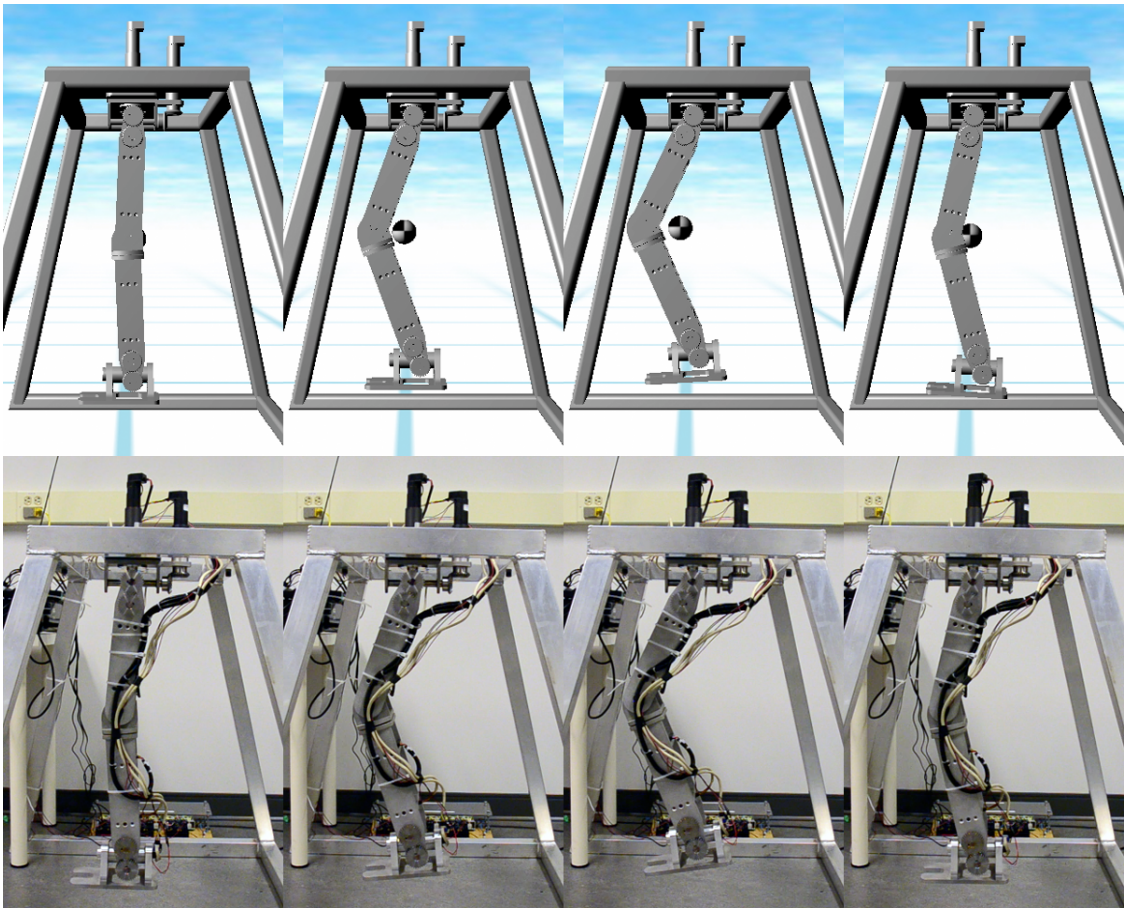


Figure 6.12: Captured frames while raising the foot in simulation and physical hardware.

Raising the Foot

In this experiment, the foot position is raised vertically, resulting in a knee bending motion. The frame captures shown in Figure 6.12 demonstrate this experiment in simulation and on the physical hardware. The first frame in the sequence is the home configuration of the 7 DOF leg at start of each experiment (i.e. at $t = 0$). The second frame represents a slightly angled configuration of the leg joints after $t_{init} = 5$ seconds. The remaining frames demonstrate the foot raising motion for this experiment. The differences in ankle pitch between the simulation and hardware are due to the discrepancies in the robots *home* position at the start of each experiment. In the simulation environment, the ankle pitch joint angle is perfectly at $q = 0$ at $t = 0$s. However, the physical robot is manually adjusted by hand to some joint angle $q = q_{init}$ which only approximates the home position. The discrepancies arise since the rotary encoders only provide the relative angular displacement of the joint from the initial joint angle q_{init} and the joint level command is obtained using the Jacobian in (6.12).

PD control was also applied on the task space level to improve performance of the motion controller. The gains were tuned in simulation to achieve the desired tracking performance before attempting the experiment on the physical hardware. The foot position tracking using $K_P = 10$ and $K_D = 0.1$ is shown in Figure 6.13 in simulation and on hardware.

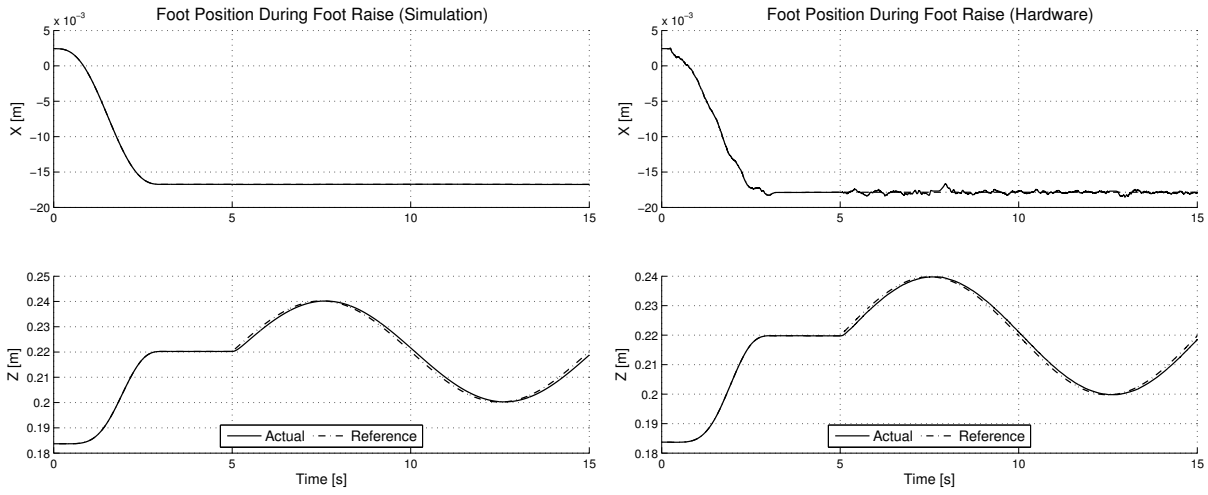


Figure 6.13: Task space trajectory tracking for bending the knee and raising the leg in simulation and hardware.

The trajectory on the physical hardware along the x direction is not as smooth as its simulated counterpart. Unmodeled backlash in the pitch joints (e.g. the knee) is the cause behind this difference. However, the PD control gains in the task space and joint space controllers are high enough to achieve the desired tracking performance.

The experimental results demonstrate the simplified version of the motion control framework with $n = 3$ DOF. The pitch joint trajectories are automatically generated from the desired swing foot height and the resulting motion bends the knee joint as the foot is raised. The joint level trajectories were tracked by the local PD controllers with individual gains tuned from Section 6.4.

Swing Foot Motion

In this experiment, the foot is commanded to track an arc-like trajectory while forward progress is made along the plane. The resulting motion is commonly used during the swing phase of the gait cycle for a bipedal robot to take a step. The frame captures shown in Figure 6.14 demonstrate this motion in simulation and on hardware. The task space

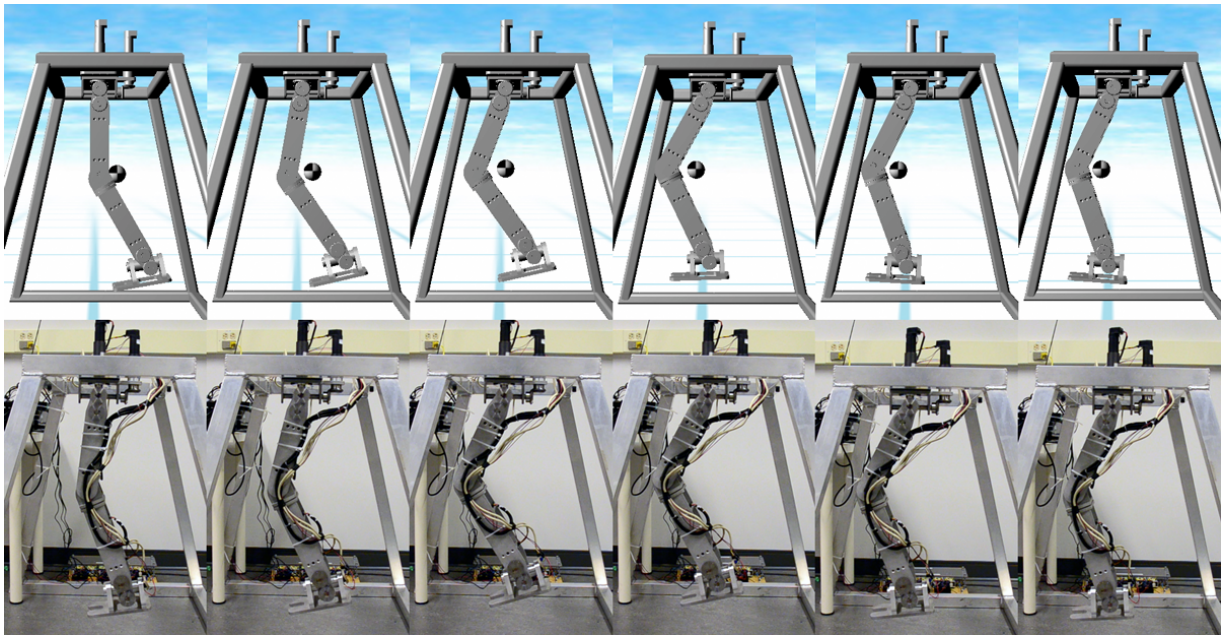


Figure 6.14: Captured frames during swing foot motion experiment in simulation and physical hardware.

control gains tuned in the previous section were reused for this experiment.

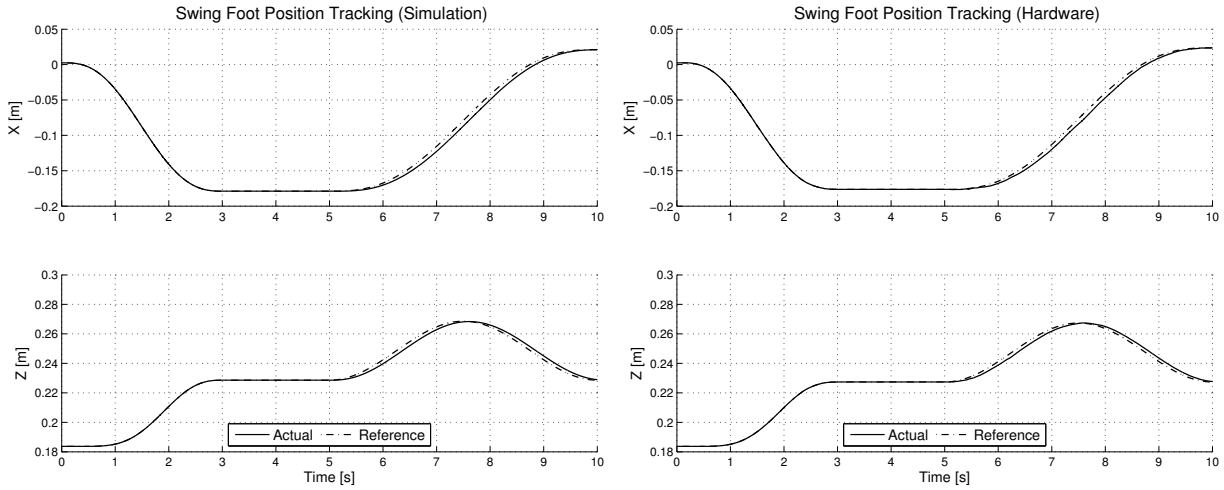


Figure 6.15: Task space trajectory tracking for swing foot motion in simulation and hardware.

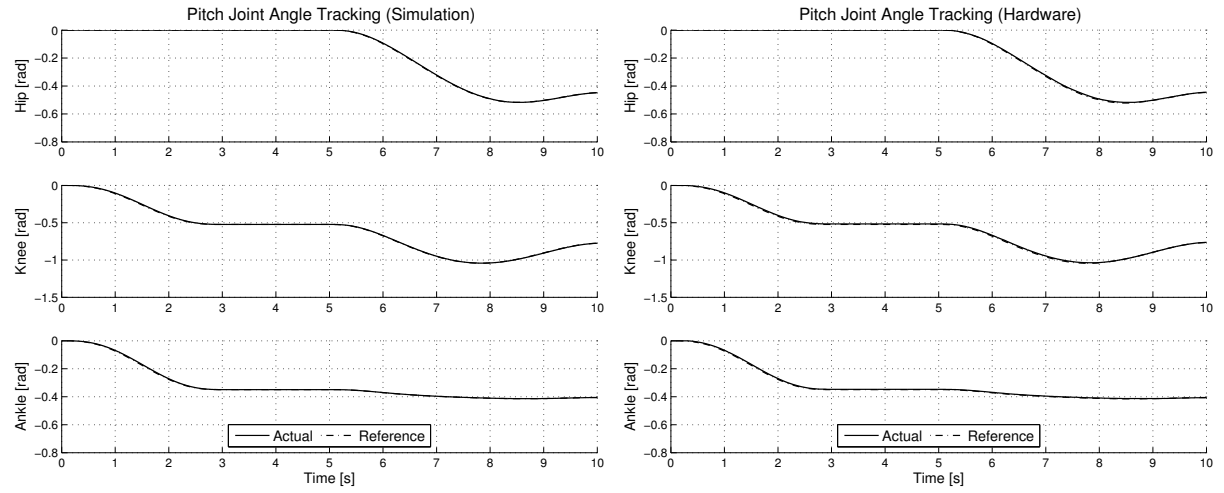


Figure 6.16: Joint space tracking of generated trajectories from higher level control during swing foot motion.

The task space tracking results for the swing foot motion is shown in Figure 6.15. Similar to the previous planar motion experiment, the hardware closely matches the simulated system with the tuned gain parameters. The joint space tracking of the command signal generated by the motion control framework is shown in Figure 6.16.

6.5.2 Whole Body Motion Control

Most common motions for a 3D bipedal robot are not restricted to the planar case. These require control over the whole body during the gait cycle. The strategy developed in Section 5.1.3 relies heavily on controlling the biped's motion through the Jacobian for the COM ($\mathbf{J_{COM}}$) and both feet ($\mathbf{J_{STAND}}$, $\mathbf{J_{SWING}}$). These are used in conjunction with the prioritized control scheme to maintain stability in the off sagittal plane and generate a forward momentum for balance recovery with the 3D FPE approach. The 7 DOF experimental platform is labeled as the swing leg for the following experiments.

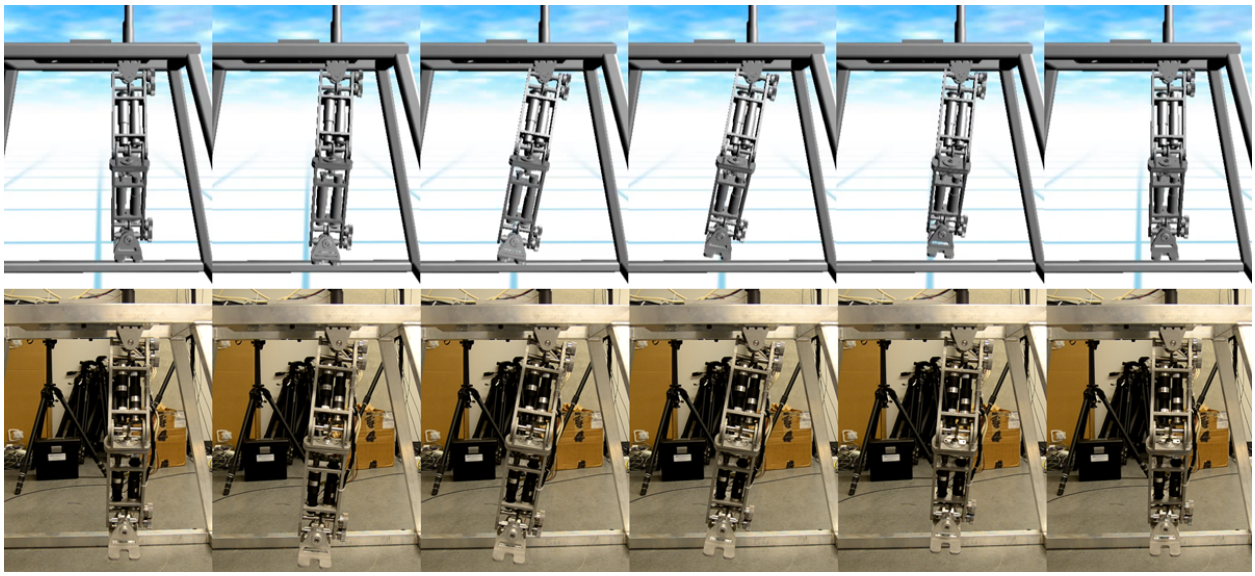


Figure 6.17: Captured frames during COM circular experiment in simulation and physical hardware.

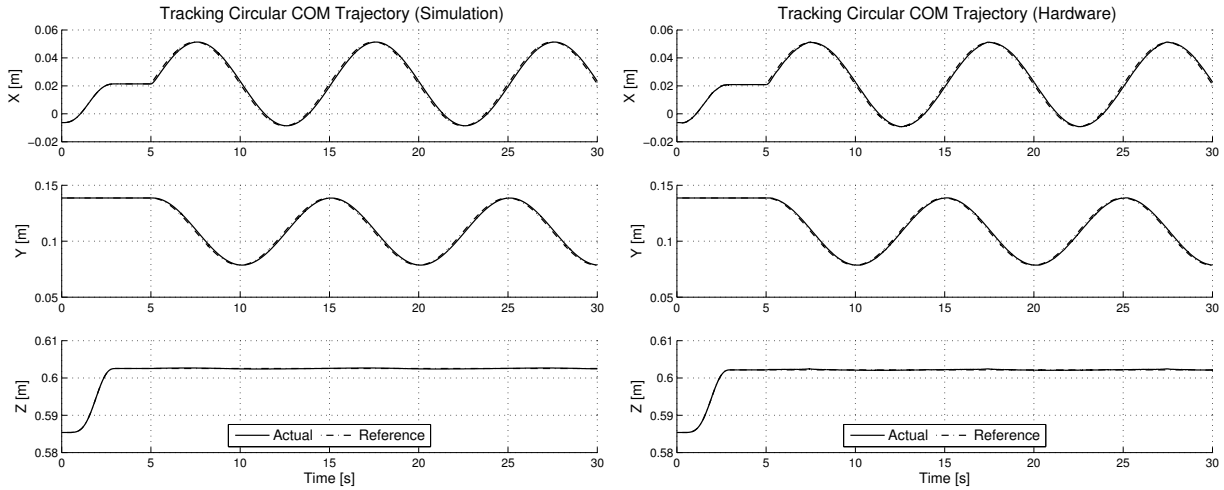


Figure 6.18: Task space tracking of circular COM motion in software and hardware.

Circling COM Motion

A circular trajectory is generated for $\mathbf{x_{COM}}$ on the xy-plane at a fixed height $z = h_{fixed}$. The height of the COM position at the end of t_{init} is selected as this fixed height. Similarly to the planar case, the joint level commands are generated by the Jacobian mapping in (6.14).

$$\dot{\mathbf{q}}_{\mathbf{ref}} = \mathbf{J}^{\#}_{\mathbf{COM}} \dot{\mathbf{x}}_{\mathbf{COM}} \tag{6.14}$$

$\dot{\mathbf{x}}_{\mathbf{COM}}$ represents the cartesian position of the foot in the world frame. The tracking results of the $\mathbf{x_{COM}}$ following the circular trajectory are shown in Figure 6.18. The initial task space control gains used in Section 6.5.1 required retuning. The tuned gain values $K_P = 14$ and $K_D = 0.5$ achieved nearly identical performance on hardware and software.

Prioritized Motion

The prioritized control scheme is used to project a low priority task ($\mathbf{J_L}$, $\dot{\mathbf{x}}_{\mathbf{L}}$) onto the null space of a higher priority task ($\mathbf{J_H}$, $\dot{\mathbf{x}}_{\mathbf{H}}$). This approach was used in Section 5.1.3 to maintain the $\mathbf{x_{COM}}$ position at the center of the support foot ($\mathbf{x_{STAND}}$) while the swing foot is lifted from the ground. A side stepping motion is generated for $\mathbf{x_{SWING}}$ to follow

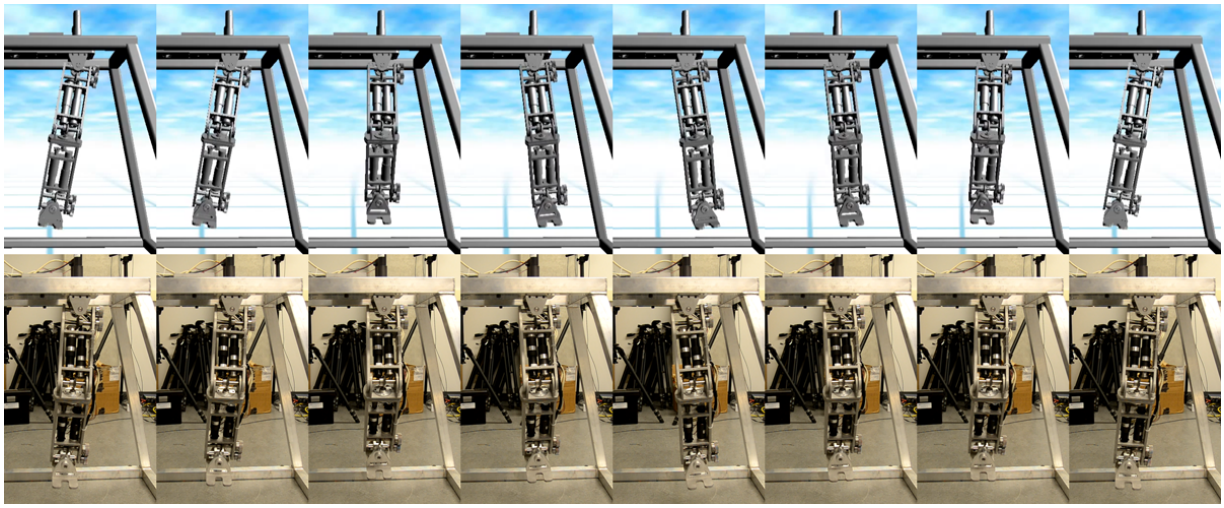


Figure 6.19: Captured frames during prioritized motion experiment in simulation and physical hardware.

an arc-like trajectory along the biped's frontal (yz) plane as the high priority task for this experiment. The low priority task is selected to constrain the $\mathbf{x_{COM}}$ motion parallel to the frontal plane. That is, $\mathbf{x_{COM}}_x$ remains constant while $(\mathbf{x_{COM}}_y, \mathbf{x_{COM}}_z)$ move freely during the side stepping motion. Frame captures taken during this experiment are shown in Figure 6.19. The joint level commands consistent with the prioritization in (6.15) are computed using (5.2).

$$\begin{aligned} \mathbf{J_H} &= \left[\mathbf{J_{SWING}}\right] \dot{\mathbf{x}}_\mathbf{H} = \left[\dot{\mathbf{x}}_\mathbf{SWING}\right] \\ \mathbf{J_L} &= \left[\mathbf{J_{COM}}_x\right] \quad \dot{\mathbf{x}}_\mathbf{L} = \left[\dot{\mathbf{x}}_{\mathbf{COM}x}\right] \end{aligned} \tag{6.15}$$

$\mathbf{J_{COM}}_x$ and $\dot{\mathbf{x}}_{\mathbf{COM}x}$ represent the x-component (i.e. the first row) of the COM Jacobian and task space error $\Delta\mathbf{x_{COM}}$. The prioritized tracking results for this experiment are shown in Figure 6.20. The tracking performance of the high priority tasks are almost identical in both cases. However, the low priority tracking performance is noisier on the hardware when compared to the simulation results.

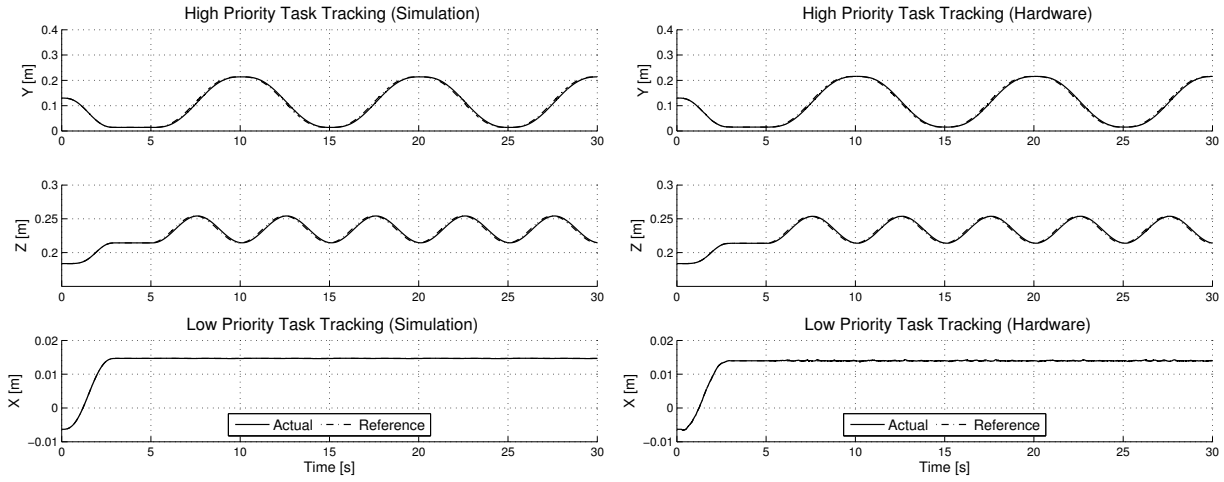


Figure 6.20: Tracking high and low priority task space trajectories in simulation and hardware.

6.6 Summary

Validating control algorithms on physical hardware is important due to modeling imperfections in the simulation environment. The physical 14 DOF bipedal robot developed in earlier chapters was used for experimental validation. The custom control electronics solution was developed by Quanser Inc for the bipedal robot. Unfortunately, the complete solution was not ready in time for experimental validation. An interim solution was used to control an assembled 7 DOF leg for experimental validation in this section.

For the physical hardware, DC motors are controlled by a voltage signal v_m. A second order system was used to model the actuator dynamics and reformulate the existing controllers which controlled the link-side torques τ_k. Independent joint control decouples each joint for individual control. Since this approach does not account for the coupled dynamics of the overall system, the link-side torques are treated as a disturbance to the actuator dynamics model. The effective motor inertia is reformulated to include the link inertia seen by the joint. A generic RNE-based algorithm was used to extract the mass matrix of the system at every time step to compute the effective motor inertia.

The HIL architecture was developed using Quanser's QUARC toolbox, DAQ board and a voltage amplifier. The controller code developed in Simulink was targeted to execute on the PC. The DAQ board allowed the PC-based controller to communicate with the physical

hardware through analog and digital I/O. Quadrature decoding was on the encoder output to obtain the link-side joint angles on physical hardware for closed loop feedback control. Parallel simulation models were developed for experimental validation to target the physical hardware or the simulated actuator dynamics.

The parallel models were used to demonstrate key components of the control approach developed in this thesis on physical hardware. Single DOF validation was used to individually tune the PD controller gains at the joint level. Once the desired tracking performance was achieved, the task space motion control framework used to generate the joint level commands was validated. Starting with a simple planar motion control problem, the foot position was controlled in the task space to achieve knee bending and an arc-like swing foot motion. Joint level command signals were generated for the local PD controllers using the simplified motion control framework. PD control was also applied at the task space level to improve tracking performance of the motion controller. Finally, the complete motion control framework was demonstrated in simulation and on the physical hardware through the COM tracking experiment. By using all (7) actuated DOF in the leg, the controller was able to sufficiently track a task space trajectory for the COM position. The prioritized task space control scheme was also demonstrated by restricting the $\mathbf{x_{COM}}$ motion to a plane in the null space of the higher priority constraint of tracking an arc-like trajectory for side-to-side stepping.

Chapter 7

Conclusions

This thesis presented a novel walking control strategy for on-line gait synthesis and the development of a 3D 14 DOF bipedal robot for experimental validation.

The electromechanical design specifications for the bipedal robot were derived using motion captured human gait data. Electric motors used in the drivetrain were selected based on these specifications and DC motor characteristics. The chassis and power transmission system were designed in CAD to improve the controllability during walking. A rapid prototyping toolchain was developed to streamline the iterative design of complex multibody systems. Full dynamic simulations and real-time 3D visualizations were automatically generated for the biped using this toolchain. It was also used to revise the physical design prior to manufacturing.

The FPE algorithm was extended to form complete gait cycles for 3D bipedal robots. The proposed algorithm selects a 2D plane in a chosen direction of motion and generates forward momentum along the plane. A trajectory generation scheme was developed for the COM and swing foot to maintain balance in the off-sagittal plane while generating the forward momentum. By solving the 2D FPE equation along the selected plane, a 3D biped was shown to regain stability by taking a step in the chosen direction of motion. A whole body motion control framework coupled with a state machine was used to form dynamically stable gait cycles. The efficacy of the proposed algorithm was demonstrated in dynamic simulations for side stepping and forward walking.

Experimental validation on the physical hardware was presented. A HIL architecture was used to develop parallel controller models which can target the simulated or physical bipedal robot. The actuator dynamics and motion control framework were validated through the parallel models for single joints and a complete leg.

7.1 Future Work

Directions for future work are divided into software and hardware. First, the simulation environment can be improved for more accurate modeling of ground reaction force. The parallel models used for experimental validation on hardware make it more valuable to have accurate simulations. Secondly, the complete walking control and gait synthesis strategy must be implemented on the hardware for more extensive validation.

7.1.1 Simulations

One of the most difficult challenges in setting up the dynamic simulations in Chapter 5 was the contact model. The initial spring-damper model was replaced with the more complex Hunt and Crossley model to eliminate discontinuities at the contact point. The difficult task was to appropriately tune the parameters to model a stiff ground contact. This was particularly important due to the change of weight loading during the double to single support transitions. Poorly tuned contact models (soft ground) resulted in oscillations as the stance foot attempted to support the entire weight of the biped. This often destabilized the biped during the **LIFT** and **SWING** states in the off-sagittal plane.

Raising the contact parameters to model a stiff ground resulted in singularities for large time steps using a fixed step solver. These singularities could often be avoided by selecting a smaller time step. However, this drastically increased the simulation run times making it difficult to iteratively adjust the control strategy and observe the resulting behaviour in dynamic simulations. The use of variable step solvers can be explored as an option to reduce simulation runtime while providing a stiffer ground with much higher contact model constants. The key challenge here would be to verify the modeling accuracy with the new solver.

Another direction for future work in simulation is to analyze turning with each step. The simulations can be used to determine the minimum stable turning radius and evaluate the turning performance of the proposed walking control strategy.

7.1.2 Experiments

The experimental validation presented in Chapter 6 was limited to controlling the motion of a single leg due to hardware constraints. While the experiments validate the simulation models for key components of the proposed strategy, the complete algorithm must be implemented on the physical 14 DOF bipedal robot for more extensive validation.

It is difficult to accurately model some dynamic forces in simulation (e.g. contact, friction), while other effects were left completely unmodeled (e.g. backlash in the geartrain). The specific difficulties with contact modeling mentioned in Section 7.1.1 reinforce the need for experimental validation of a leg in contact with the ground. While it may be easier to control the biped under single support due to a physically stiff ground, the torque loading on the motors can be different from the simulation environment. More extensive validation can also provide a source for tuning the contact model parameters in simulation to improve the modeling accuracy.

References

[1] M. Vukobratović and B. Borovac, "Zero-moment point thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[2] S. Kajita and K. Tani, "Adaptive gait control of a biped robot based on realtime sensing of the ground profile," *Autonomous Robots*, vol. 4, no. 3, pp. 297–305, 1997.

[3] T. Sugihara, Y. Nakamura, and H. Inoue, "Real-time humanoid motion generation through ZMP manipulation based on inverted pendulum control," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 2, pp. 1404–1409, IEEE, 2002.

[4] T. McGeer, "Passive dynamic walking," *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 1990.

[5] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

[6] M. Spong *et al.*, "Passivity based control of the compass gait biped," in *Proc. of IFAC World Congress, Beijing, China*, 1999.

[7] D. L. Wight, *A Foot Placement Strategy for Robust Bipedal Gait Control*. PhD thesis, University of Waterloo, 2008.

[8] D. Wight, E. Kubica, and D. Wang, "Introduction of the foot placement estimator: A dynamic measure of balance for bipedal robotics," *Journal of computational and nonlinear dynamics*, vol. 3, no. 1, 2008.

[9] A. Goswami, B. Espiau, and A. Keramane, "Limit cycles and their stability in a passive bipedal gait," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 1, pp. 246–251, IEEE, 1996.

[10] F. Asano, M. Yamakita, and K. Furuta, "Virtual passive dynamic walking and energy-based control laws," in *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 2, pp. 1149–1154, IEEE, 2000.

[11] A. Kuo, "Stabilization of lateral motion in passive dynamic walking," *The International Journal of Robotics Research*, vol. 18, no. 9, pp. 917–930, 1999.

[12] L. RYGG, "Mechanical horse," Feb. 14 1893. US Patent 491,927.

[13] G. PALLIS, "Fallis," Jan. 17 1888. US Patent 376,588.

[14] M. Vukobratovic and D. Juricic, "Contribution to the synthesis of biped gait," *Biomedical Engineering, IEEE Transactions on*, no. 1, pp. 1–6, 1969.

[15] I. Kato, "The hydraulically powered biped walking machine with a high carrying capacity," in *Proc. of the 4th Int. Symposium on External Control of Human Extremities, Dubrovnik*, pp. 410–421, 1972.

[16] I. Mizuuchi, Y. Nakanishi, Y. Sodeyama, Y. Namiki, T. Nishino, N. Muramatsu, J. Urata, K. Hongo, T. Yoshikai, and M. Inaba, "An advanced musculoskeletal humanoid kojiro," in *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pp. 294–299, IEEE, 2007.

[17] M. Osada, N. Ito, Y. Nakanishi, and M. Inaba, "Realization of flexible motion by musculoskeletal humanoid "Kojiro" with add-on nonlinear spring units," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pp. 174–179, IEEE, 2010.

[18] S. Tzafestas, M. Raibert, and C. Tzafestas, "Robust sliding-mode control applied to a 5-link biped robot," *Journal of Intelligent & Robotic Systems*, vol. 15, no. 1, pp. 67–133, 1996.

[19] C. Chevallereau and P. Sardain, "Design and actuation optimization of a 4-axes biped robot for walking and running," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 4, pp. 3365–3370, IEEE, 2000.

[20] J. Pratt, C. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: An intuitive approach for bipedal locomotion," *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001.

[21] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of Honda humanoid robot," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2, pp. 1321–1326, IEEE, 1998.

[22] I. Park, J. Kim, J. Lee, and J. Oh, "Mechanical design of humanoid robot platform KHR-3 (KAIST humanoid robot 3: HUBO)," in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pp. 321–326, IEEE, 2005.

[23] K. Kaneko, K. Harada, F. Kanehiro, G. Miyamori, and K. Akachi, "Humanoid robot HRP-3," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 2471–2478, IEEE, 2008.

[24] Y. Ogura, H. Aikawa, K. Shimomura, A. Morishima, H. Lim, and A. Takanishi, "Development of a new humanoid robot WABIAN-2," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 76–81, IEEE, 2006.

[25] J. Yamaguchi, E. Soga, S. Inoue, and A. Takanishi, "Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 1, pp. 368–374, IEEE, 1999.

[26] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "The nao humanoid: a combination of performance and affordability," *CoRR, vol. abs/0807.3223*, 2008.

[27] B. Dasgupta and T. Mruthyunjaya, "The Stewart platform manipulator: a review," *Mechanism and machine theory*, vol. 35, no. 1, pp. 15–40, 2000.

[28] Y. Sugahara, A. Ohta, K. Hashimoto, H. Sunazuka, M. Kawase, C. Tanaka, H. Lim, and A. Takanishi, "Walking up and down stairs carrying a human by a biped locomotor with parallel mechanism," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 1489–1494, IEEE, 2005.

[29] C. Gosselin and J. Hamel, "The agile eye: A high-performance three-degree-of-freedom camera-orienting device," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 781–786, IEEE, 1994.

[30] J. Hofschulte, M. Seebode, and W. Gerth, "Parallel manipulator hip joint for a bipedal robot," *Climbing and Walking Robots*, pp. 601–609, 2005.

[31] S. Kajita, T. Yamaura, and A. Kobayashi, "Dynamic walking control of a biped robot along a potential energy conserving orbit," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 4, pp. 431–438, 1992.

[32] M. Morisawa, T. Yakoh, T. Murakami, and K. Ohnishi, "A comparison study between parallel and serial linked structures in biped robot system," in *Industrial Electronics Society, 2000. IECON 2000. 26th Annual Confjerence of the IEEE*, vol. 4, pp. 2614–2619, IEEE, 2000.

[33] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent ASIMO: System overview and integration," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3, pp. 2478–2483, IEEE, 2002.

[34] J. Merlet, *Parallel robots*. Springer, 2006.

[35] A. Konno, R. Sellaouti, F. Amar, and F. Ouezdou, "Design and development of the biped prototype ROBIAN," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 2, pp. 1384–1389, IEEE, 2002.

[36] R. Sellaouti and F. Ouezdou, "Design and control of a 3DOFs parallel actuated mechanism for biped application," *Mechanism and machine theory*, vol. 40, no. 12, pp. 1367–1393, 2005.

[37] S. Lohmeier, T. Buschmann, M. Schwienbacher, H. Ulbrich, and F. Pfeiffer, "Leg design for a humanoid walking robot," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pp. 536–541, IEEE, 2006.

[38] Y. Yiu, H. Cheng, Z. Xiong, G. Liu, and Z. Li, "On the dynamics of parallel manipulators," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4, pp. 3766–3771, IEEE, 2001.

[39] W. Shang, S. Cong, and Y. Zhang, "Nonlinear friction compensation of a 2-DOF planar parallel manipulator," *Mechatronics*, vol. 18, no. 7, pp. 340–346, 2008.

[40] T. McGeer, "Passive walking with knees," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pp. 1640–1645, IEEE, 1990.

[41] M. Garcia, A. Chatterjee, and A. Ruina, "Efficiency, speed, and scaling of two-dimensional passive-dynamic walking," *Dynamics and Stability of Systems*, vol. 15, no. 2, pp. 75–99, 2000.

[42] S. Collins, M. Wisse, and A. Ruina, "A three-dimensional passive-dynamic walking robot with two legs and knees," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 607–615, 2001.

[43] M. Wisse and R. van der Linde, *Delft Pneumatic Bipeds*. Springer Tracts in Advanced Robotics, Springer, 2007.

[44] M. Wisse and J. Van Frankenhuyzen, "Design and construction of mike; a 2D autonomous biped based on passive dynamic walking," in *Proceedings of international symposium of adaptive motion and animals and machines (AMAM03)*, 2003.

[45] D. Hobbelen and M. Wisse, "Ankle joints and flat feet in dynamic walking," *Climbing and Walking Robots*, pp. 787–800, 2005.

[46] D. Hobbelen, T. de Boer, and M. Wisse, "System overview of bipedal robots flame and tulip: Tailor-made for limit cycle walking," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 2486–2491, IEEE, 2008.

[47] M. Wisse, G. Feliksdal, J. Van Frankkenhuyzen, and B. Moyer, "Passive-based walking robot," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 2, pp. 52–62, 2007.

[48] B. Vanderborght, B. Verrelst, R. Van Ham, M. Van Damme, and D. Lefeber, "A pneumatic biped: experimental walking results and compliance adaptation experiments," in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pp. 44–49, IEEE, 2005.

[49] G. Bilodeau and E. Papadopoulos, "A model-based impedance control scheme for high-performance hydraulic joints," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 2, pp. 1308–1313, IEEE, 1998.

[50] S. Hyon, J. Hale, and G. Cheng, "Full-body compliant human–humanoid interaction: balancing in the presence of unknown external forces," *Robotics, IEEE Transactions on*, vol. 23, no. 5, pp. 884–898, 2007.

[51] S. Anderson, M. Wisse, C. Atkeson, J. Hodgins, G. Zeglin, and B. Moyer, "Powered bipeds based on passive dynamic principles," in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pp. 110–116, IEEE, 2005.

[52] G. Pratt and M. Williamson, "Series elastic actuators," in *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 1, pp. 399–406, IEEE, 1995.

[53] A. Kuo *et al.*, "Energetics of actively powered locomotion using the simplest walking model," *Transactions-american Society Of Mechanical Engineers Journal Of Biomechanical Engineering*, vol. 124, no. 1, pp. 113–120, 2002.

[54] D. Hobbelen and M. Wisse, "Ankle actuation for limit cycle walkers," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 709–735, 2008.

[55] D. Hobbelen and M. Wisse, "Controlling the walking speed in limit cycle walking," *The International Journal of Robotics Research*, vol. 27, no. 9, pp. 989–1005, 2008.

[56] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie, "Planning walking patterns for a biped robot," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 3, pp. 280–289, 2001.

[57] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 1, pp. 239–246, IEEE, 2001.

[58] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2, pp. 1620–1626, IEEE, 2003.

[59] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot - 1st report: Walking gait pattern generation," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 1084–1091, IEEE, 2009.

[60] D. Gouaillier and P. Blazevic, "A mechatronic platform, the Aldebaran robotics humanoid robot," in *IEEE Industrial Electronics, IECON 2006-32nd Annual Conference on*, pp. 4049–4053, IEEE, 2006.

[61] S. Kajita, M. Morisawa, K. Harada, K. Kaneko, F. Kanehiro, K. Fujiwara, and H. Hirukawa, "Biped walking pattern generator allowing auxiliary zmp control," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 2993–2999, IEEE, 2006.

[62] F. Asano, Z. Luo, and M. Yamakita, "Some extensions of passive walking formula to active biped robots," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4, pp. 3797–3802, IEEE, 2004.

[63] F. Asano, Z. Luo, and M. Yamakita, "Unification of dynamic gait generation methods via variable virtual gravity and its control performance analysis," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 4, pp. 3865–3870, IEEE, 2004.

[64] J. Pratt, *Virtual model control of a biped walking robot.* PhD thesis, Massachusetts Institute of Technology, 1995.

[65] J. Pratt and G. Pratt, "Intuitive control of a planar bipedal walking robot," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 3, pp. 2014–2021, IEEE, 1998.

[66] J. Pratt, C. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: An intuitive approach for bipedal locomotion," *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001.

[67] B. Stephens and C. Atkeson, "Dynamic balance force control for compliant humanoid robots," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 1248–1255, IEEE, 2010.

[68] S. Hyon, J. Hale, and G. Cheng, "Full-body compliant human–humanoid interaction: balancing in the presence of unknown external forces," *Robotics, IEEE Transactions on*, vol. 23, no. 5, pp. 884–898, 2007.

[69] C. Ott, C. Baumgartner, J. Mayr, M. Fuchs, R. Burger, D. Lee, O. Eiberger, A. Albu-Schaffer, M. Grebenstein, and G. Hirzinger, "Development of a biped robot with torque controlled joints," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pp. 167–173, IEEE, 2010.

[70] D. Katić and M. Vukobratović, "Survey of intelligent control techniques for humanoid robots," *Journal of Intelligent & Robotic Systems*, vol. 37, no. 2, pp. 117–141, 2003.

[71] W. Miller III, "Real-time neural network control of a biped walking robot," *Control Systems, IEEE*, vol. 14, no. 1, pp. 41–48, 1994.

[72] A. Kun and W. Miller III, "Adaptive dynamic balance of a biped robot using neural networks," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 1, pp. 240–245, IEEE, 1996.

[73] G. Taga, Y. Yamaguchi, and H. Shimizu, "Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment," *Biological cybernetics*, vol. 65, no. 3, pp. 147–159, 1991.

[74] G. Taga, "A model of the neuro-musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance," *Biological Cybernetics*, vol. 78, no. 1, pp. 9–17, 1998.

[75] S. Miyakoshi, G. Taga, Y. Kuniyoshi, and A. Nagakubo, "Three dimensional bipedal stepping motion using neural oscillators-towards humanoid motion in the real world," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 1, pp. 84–89, IEEE, 1998.

[76] A. Ijspeert, "2008 Special Issue: Central pattern generators for locomotion control in animals and robots: A review," *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.

[77] J. Morimoto, G. Endo, J. Nakanishi, S. Hyon, G. Cheng, D. Bentivegna, and C. Atkeson, "Modulation of simple sinusoidal patterns by a coupled oscillator model for biped walking," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 1579–1584, IEEE, 2006.

[78] L. Righetti and A. Ijspeert, "Programmable central pattern generators: an application to biped locomotion control," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 1585–1590, IEEE, 2006.

[79] S. Aoi and K. Tsuchiya, "Locomotion control of a biped robot using nonlinear oscillators," *Autonomous Robots*, vol. 19, no. 3, pp. 219–232, 2005.

[80] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pp. 200–207, IEEE, 2006.

[81] J. Englsberger, C. Ott, M. Roa, A. Albu-Schaffer, and G. Hirzinger, "Bipedal walking control based on Capture Point dynamics," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 4420–4427, IEEE, 2011.

[82] M. Krause, J. Englsberger, P. Wieber, and C. Ott, "Stabilization of the Capture Point Dynamics for Bipedal Walking Based on Model Predictive Control," in *Robot Control*, vol. 10, pp. 165–171, 2012.

[83] P. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pp. 137–142, IEEE, 2006.

[84] T. De Boer, *Foot placement in robotic bipedal locomotion*. PhD thesis, Delft University of Technology, 2012.

[85] T. Koolen, T. De Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[86] J. Pratt and B. Krupp, "Design of a bipedal walking robot," in *SPIE Defense and Security Symposium*, pp. 69621F–69621F, International Society for Optics and Photonics, 2008.

[87] J. Pratt, T. Koolen, T. De Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus, "Capturability-based analysis and control of legged locomotion, Part 2: Application to M2V2, a lower-body humanoid," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1117–1133, 2012.

[88] M. Millard, *Mechanics and Control of Human Balance*. PhD thesis, University of Waterloo, 2011.

[89] B. Perrin, C. Chevallereau, and C. Verdier, "Calculation of the direct dynamic model of walking robots: Comparison between two methods," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 2, pp. 1088–1093, IEEE, 1997.

[90] S. Kajita, *Humanoid Robots*. Ohmsha, 2005.

[91] M. van der Krogt and J. Harlaar, "Gait Analysis Tutorial — Dynamic Walking 2008." `http://dynamicwalking.org/dw2008/?q=node/22`, 2008. [Online; Accessed 02-February-2011].

[92] J. Harlaar, "BodyMech: A Matlab based open source package for 3D kinematic analysis." `http://www.bodymech.nl/`, 2006. [Online; Accessed 03-February-2011].

[93] Faulhaber Group, "Micromo: Small DC Motor Drive Systems."

[94] D. A. Winter, *Biomechanics and Motor Control of Human Gait: Normal, Elderly and Pathological*. Waterloo Biomechanics, 1991.

[95] J. Kim, S. Park, I. Park, and J. Oh, "Development of a humanoid biped walking robot platform khr-1-initial design and its performance evaluation," *feedback*, vol. 1, p. 12, 2002.

[96] J. Kim, I. Park, J. Lee, M. Kim, B. Cho, and J. Oh, "System design and dynamic walking of humanoid robot khr-2," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 1431–1436, IEEE, 2005.

[97] P. Sardain, M. Rostami, and G. Bessonnet, "An anthropomorphic biped robot: dynamic concepts and technological design," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 28, no. 6, pp. 823–838, 1998.

[98] D. W. S. Choudhury and D. Kulić, "Rapid Prototyping Toolchain for Humanoid Robotics Applications," in *IEEE International Conference on Humanoid Robots*, 2012. Accepted for Publication.

[99] C. Paul and J. Bongard, "The road less travelled: Morphology in the optimization of biped robot locomotion," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 1, pp. 226–232, IEEE, 2001.

[100] D. Wollherr, M. Hardt, M. Buss, and O. von Stryk, "Actuator selection and hardware realization of a small and fast-moving, autonomous humanoid robot," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3, pp. 2491–2496, IEEE, 2002.

[101] T. Ravichandran, G. Heppler, and D. Wang, "Task-Based Optimal Simultaneous Manipulator/Controller Design Using Evolutionary Algorithms," in *Dynamics and Control of Systems and Structures in Space, 2004 6th International Conference on*, pp. 707–716, Cranfield University Press, 2004.

[102] N. Koenig and A. Howard, "Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2149–2154, 2004.

[103] O. Michel, "Webots: Professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 40–43, 2004.

[104] F. Kanehiro, H. Hirukawa, and S. Kajita, "OpenHRP: Open architecture humanoid robotics platform," *The International Journal of Robotics Research*, vol. 23, no. 2, pp. 155–165, 2004.

[105] R. Ponticelli and M. Armada, "Vrsilo2: dynamic simulation system for the biped robot silo2," in *Proceedings of the 9th International Conference on Climbing and Walking Robots, Brussels*, 2006.

[106] T. Reichenbach, "A dynamic simulator for humanoid robots," *Artificial Life and Robotics*, vol. 13, no. 2, pp. 561–565, 2009.

[107] G. Medrano-Cerda, H. Dallali, M. Brown, N. Tsagarakis, and D. Caldwell, "Modelling and simulation of the locomotion of humanoid robots," in *UK Automatic Control Conference, Coventry, 7-10 September*, 2010.

[108] D. Baraff, "Linear-time dynamics using Lagrange multipliers," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 137–146, ACM, 1996.

[109] D. ROSENTHAL and M. SHERMAN, "High performance multibody simulations via symbolic equation manipulation and Kane's method," *Journal of the Astronautical Sciences*, vol. 34, no. 3, pp. 223–239, 1986.

[110] C. Paredis, A. Diaz-Calderon, R. Sinha, and P. Khosla, "Composable models for simulation-based design," *Engineering with Computers*, vol. 17, no. 2, pp. 112–128, 2001.

[111] MapleSoft Inc, "MapleSim: High-Performance Physical Modeling."

[112] MathWorks Inc, "SimMechanics: Multibody Simulations for MATLAB and Simulink."

[113] K. Hunt and F. Crossley, "Coefficient of restitution interpreted as damping in vibroimpact," *ASME Journal of Applied Mechanics*, 1975.

[114] G. Gilardi and I. Sharf, "Literature survey of contact dynamics modelling," *Mechanism and machine theory*, vol. 37, no. 10, pp. 1213–1239, 2002.

[115] M. Spong and M. Vidyasagar, *Robot dynamics and control*. John Wiley & Sons, 2008.

[116] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. Springer, 2001.

[117] J. Pratt and B. Krupp, "Design of a bipedal walking robot," in *SPIE Defense and Security Symposium*, pp. 69621F–69621F, International Society for Optics and Photonics, 2008.

[118] C. Ott, M. Roa, and G. Hirzinger, "Posture and balance control for biped robots based on contact force optimization," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pp. 26–33, IEEE, 2011.

[119] Quanser Inc, "QuaRC: Control Simulation Software."

[120] Dassault Systèmes, "SolidWorks: 3D CAD Design Software."

[121] D. Wight, E. Kubica, and D. Wang, "Introduction of the foot placement estimator: A dynamic measure of balance for bipedal robotics," *Journal of computational and nonlinear dynamics*, vol. 3, no. 1, 2008.

[122] T. Assman, "Balance recovery of a planar biped using the Foot Placement Estimator,"

[123] H. Hirukawa, "Walking biped humanoids that perform manual labour," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 65–77, 2007.

[124] T. Geng, B. Porr, and B. Florentinwörgötter, "A reflexive neural network for dynamic biped walking control," *Neural Computation*, vol. 18, no. 5, pp. 1156–1196, 2006.

[125] L. Jalics, H. Hemami, and Y. Zheng, "Pattern generation using coupled oscillators for robotic and biorobotic adaptive periodic movement," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 1, pp. 179–184, IEEE, 1997.

[126] M. Cao and A. Kawamura, "A design method of neural oscillatory networks for generation of humanoid biped walking patterns," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 3, pp. 2357–2362, IEEE, 1998.

[127] A. Sano and J. Furusho, "Realization of natural dynamic walking using the angular momentum information," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pp. 1476–1481, IEEE, 1990.

[128] E. Westervelt, J. Grizzle, and C. Canudas de Wit, "Switching and PI control of walking motions of planar biped walkers," *Automatic Control, IEEE Transactions on*, vol. 48, no. 2, pp. 308–312, 2003.

[129] T. Aoyama, K. Sekiyama, Y. Hasegawa, and T. Fukuda, "Experimental verification of 3D bipedal walking based on Passive Dynamic Autonomous Control," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 1308–1313, IEEE, 2009.

[130] F. Asano and Z. Luo, "Pseudo virtual passive dynamic walking and effect of upper body as counterweight," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 2934–2939, IEEE, 2008.

[131] F. Asano and Z. Luo, "Dynamic analyses of underactuated virtual passive dynamic walking," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3210–3217, IEEE, 2007.

[132] H. Sasaki and M. Yamakita, "Efficient walking control of robot with torso based on passive dynamic walking," in *Mechatronics, ICM2007 4th IEEE International Conference on*, pp. 1–5, IEEE, 2007.

[133] F. Asano, Z. Luo, and M. Yamakita, "Biped gait generation and control based on a unified property of passive dynamic walking," *Robotics, IEEE Transactions on*, vol. 21, no. 4, pp. 754–762, 2005.

[134] M. Spong and F. Bullo, "Controlled symmetries and passive walking," *Automatic Control, IEEE Transactions on*, vol. 50, no. 7, pp. 1025–1031, 2005.

[135] F. Asano, M. Yamakita, N. Kamamichi, and Z. Luo, "A novel gait generation for biped walking robots based on mechanical energy constraint," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 3, pp. 565–573, 2004.

[136] R. Tedrake, T. Zhang, M. Fong, and H. Seung, "Actuating a simple 3D passive dynamic walker," vol. 5, pp. 4656–4661, 2004.

[137] H. Ohta, M. Yamakita, and K. Furuta, "From passive to active dynamic walking," *International Journal of Robust and Nonlinear Control*, vol. 11, no. 3, pp. 287–303, 2001.

[138] J. Samin and P. Fisette, *Symbolic Modeling of Multibody Systems*. Solid mechanics and its applications, Kluwer Academic Publishers, 2003.

[139] MathWorks Inc, “MATLAB - The Language of Technical Computing.”

Appendices

Appendix A

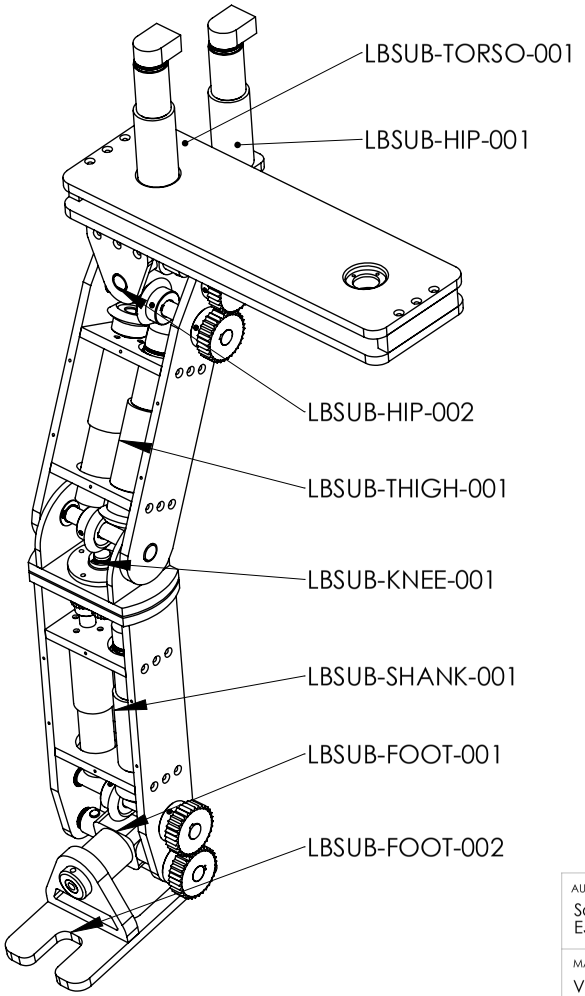
Mechanical CAD Drawings

The following drawings detail the complete mechanical design of a 14 DOF bipedal robot and supporting frame developed for experimental validation.

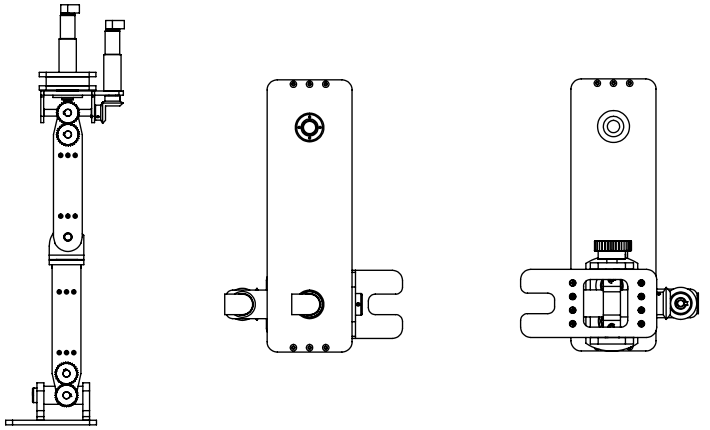
A.1 Bipedal Robot

The 14 DOF bipedal robot was designed with DC motors from Micromo[1] and drivetrain components from Misumi USA[2].

[1] http://www.micromo.com
[2] http://www.misumiusa.com



ITEM NO.	PART NUMBER	Free/QTY.
1	LBSUB-TORSO-001	1
2	LBSUB-HIP-001	2
3	LBSUB-HIP-002	2
4	LBSUB-THIGH-001	2
5	LBSUB-KNEE-001	2
6	LBSUB-SHANK-001	2
7	LBSUB-FOOT-001	2
8	LBSUB-FOOT-002	2



University of Waterloo - Adaptive Systems Laboratory

AUTHOR
Safwan Choudhury
E5 5129 (ext 31475)

TITLE:
Assembly Reference

MATERIAL
Various Materials

SIZE
A

ASSEMBLY NO.
ASLHP-LBHALF

REV
4.4

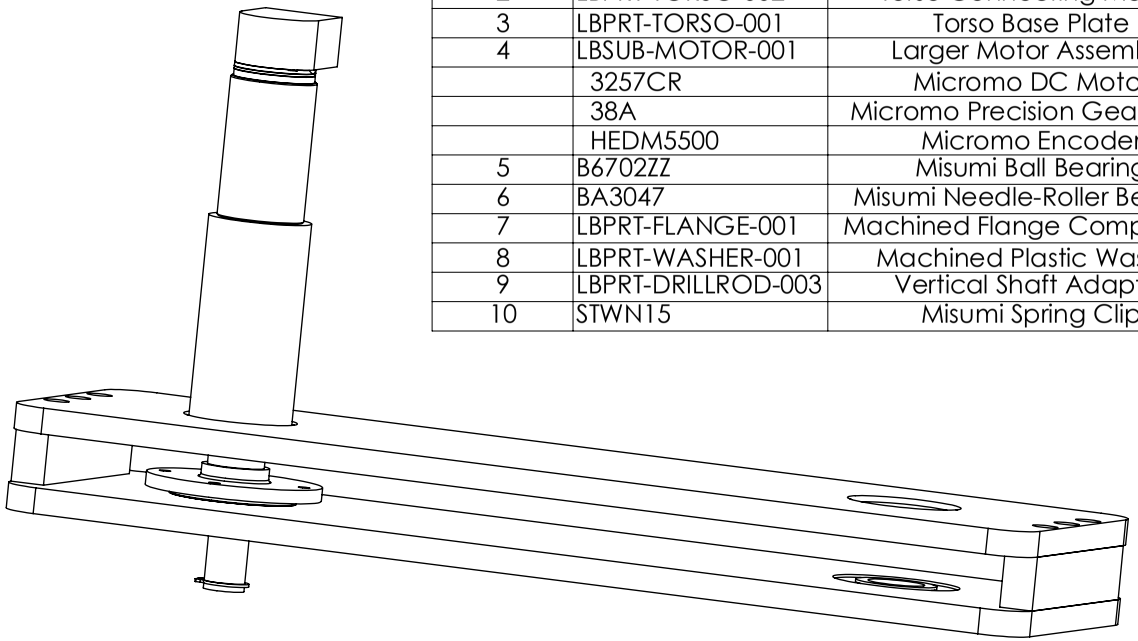
SCALE: 1:5

WEIGHT: 16kg

SHEET 1 OF 1

5 4 3 2 1

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	LBPRT-TORSO-001	Torso Base Plate	1
2	LBPRT-TORSO-002	Torso Connecting Mount	2
3	LBPRT-TORSO-001	Torso Base Plate	1
4	LBSUB-MOTOR-001	Larger Motor Assembly	1
	3257CR	Micromo DC Motor	1
	38A	Micromo Precision Gearhead	1
	HEDM5500	Micromo Encoder	1
5	B6702ZZ	Misumi Ball Bearing	1
6	BA3047	Misumi Needle-Roller Bearing	1
7	LBPRT-FLANGE-001	Machined Flange Component	1
8	LBPRT-WASHER-001	Machined Plastic Washer	1
9	LBPRT-DRILLROD-003	Vertical Shaft Adapter	1
10	STWN15	Misumi Spring Clip	1



University of Waterloo - Adaptive Systems Laboratory

AUTHOR
Safwan Choudhury
E5 5129 (ext 31475)

TITLE:
Torso Subassembly Reference

MATERIAL
Aluminum 6061

SIZE
A

ASSEMBLY NO.
LBSUB-TORSO-001

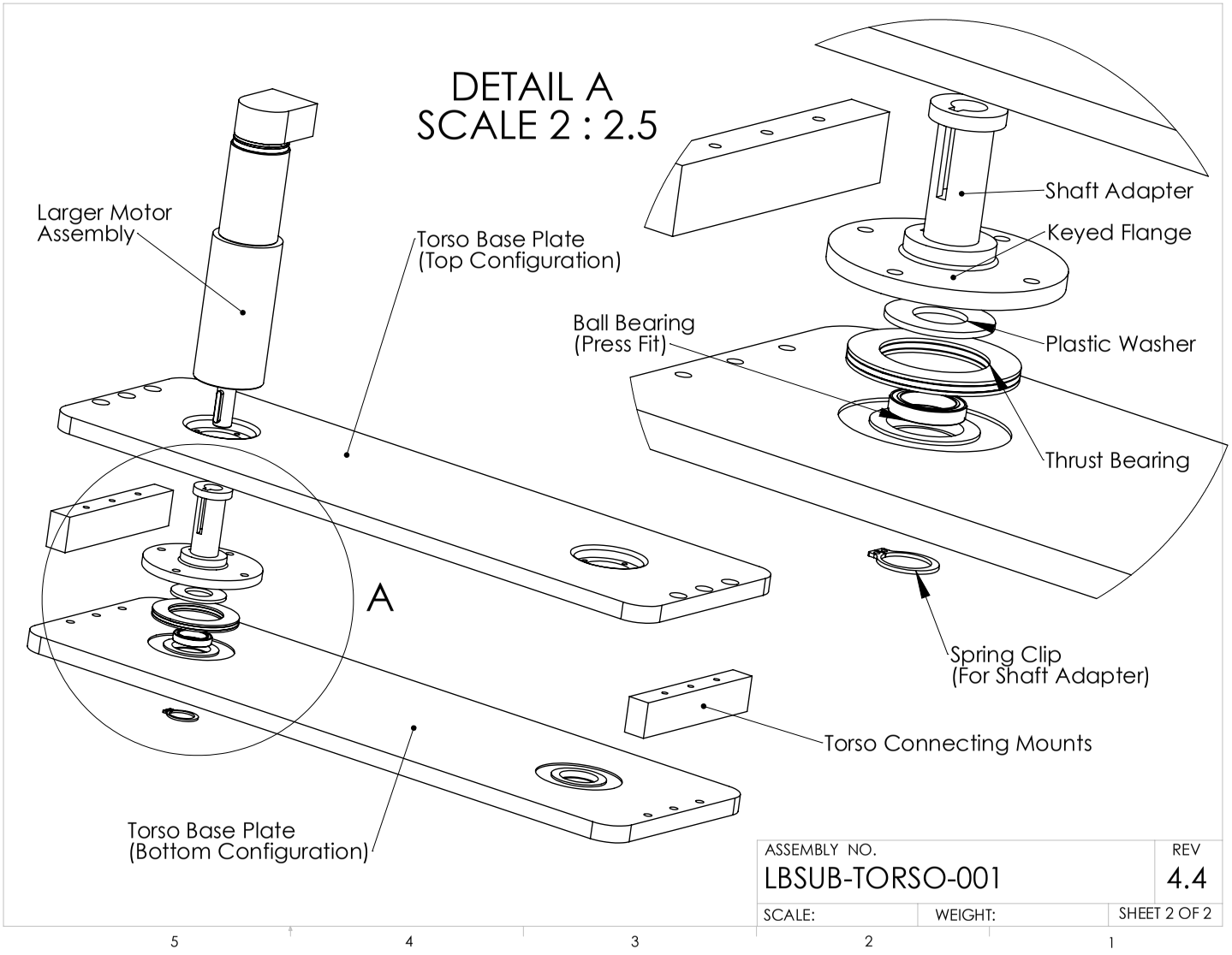
REV
4.4

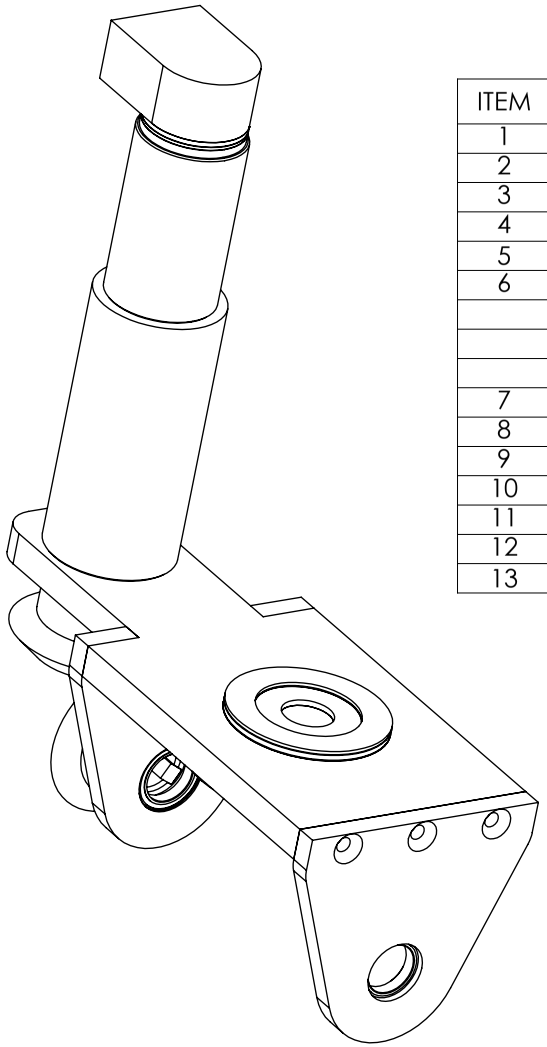
SCALE: 1:2.5

WEIGHT: 3.5kg

SHEET 1 OF 2

5 4 3 2 1


DETAIL A
SCALE 2 : 2.5
Larger Motor Assembly
Torso Base Plate (Top Configuration)
Shaft Adapter
Keyed Flange
Ball Bearing (Press Fit)
Plastic Washer
Thrust Bearing
A
Spring Clip (For Shaft Adapter)
Torso Connecting Mounts
Torso Base Plate (Bottom Configuration)
ASSEMBLY NO.
LBSUB-TORSO-001
REV
4.4
SCALE:
WEIGHT:
SHEET 2 OF 2
5
4
3
2
1



ITEM	PART NUMBER	DESCRIPTION	QTY.
1	LBPRT-HIP-001	Hip Top Plate	1
2	LBPRT-HIP-002	Hip Front/Back Plates	1
3	LBPRT-HIP-002	Hip Front/Back Plates	1
4	LBPRT-KEY-002	5mm x 5mm Key	1
5	KGEASK15-3030-15	Misumi Bevel Gear	2
6	LBSUB-MOTOR-001	Larger Motor Assembly	1
	3257CR	Micromo DC Motor	1
	38A	Micromo Precision Gearhead	1
	HEDM5500	Micromo Encoder	1
7	LBPRT-BUSHING-001	Keyed Motor Bushing	1
8	LBPRT-KEY-002	5mm x 5mm Key	1
9	B6702ZZ	Misumi Ball Bearing	2
10	BA3047	Misumi Needle-Roller Bearing	1
11	LBPRT-KEY-001	3mm x 3mm Key	1
12	LBPRT-FLANGE-001	Machined Flange Component	1
13	LBPRT-WASHER-001	Machined Plastic Washer	1

University of Waterloo - Adaptive Systems Laboratory

AUTHOR
Safwan Choudhury
E5 5129 (ext 31475)

TITLE:
Hip (1) Subassembly Reference

MATERIAL
Aluminum 6061

SIZE
A

ASSEMBLY NO.
LBSUB-HIP-001

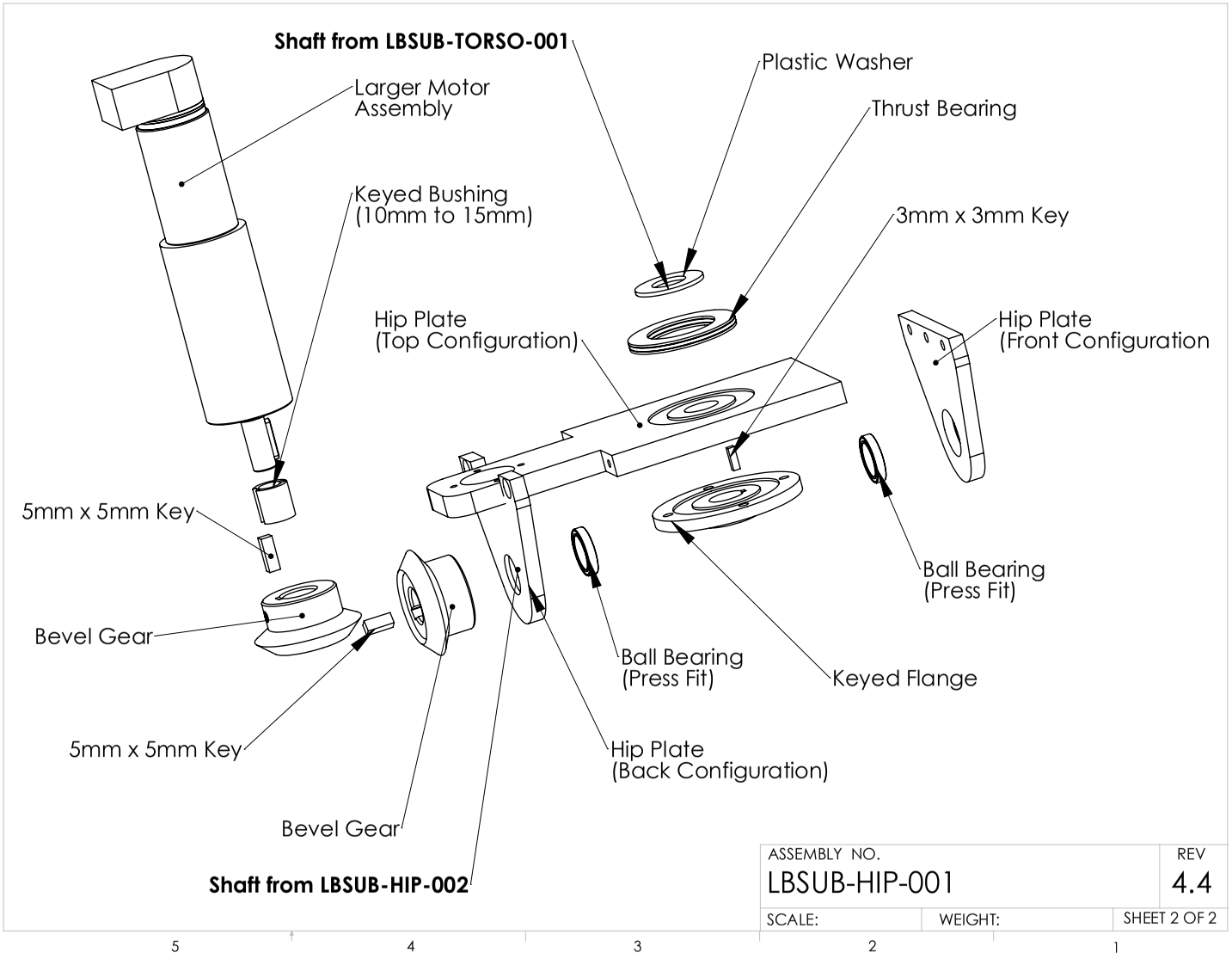
REV
4.4

SCALE: 1:1.5

WEIGHT: 1.5kg

SHEET 1 OF 2

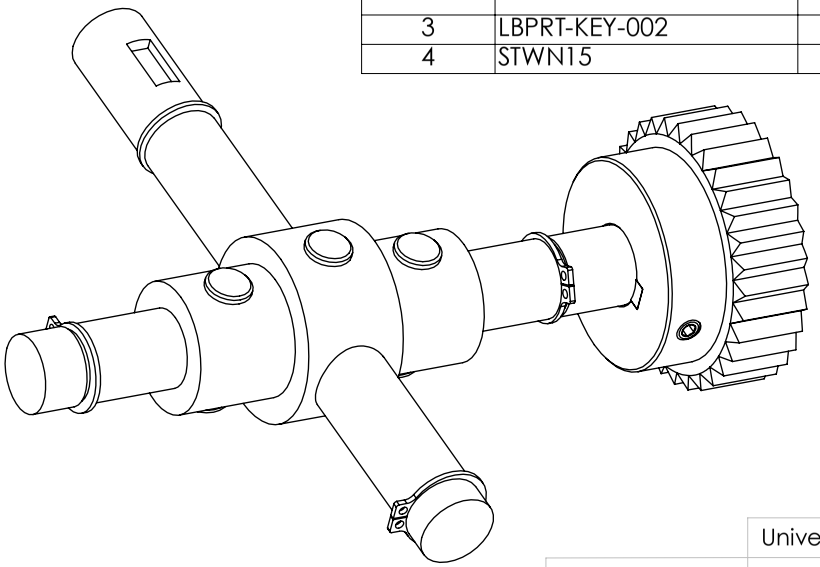
5 4 3 2 1



ASSEMBLY NO.		REV
LBSUB-HIP-001		4.4
SCALE:	WEIGHT:	SHEET 2 OF 2

5 4 3 2 1

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	LBSUB-CROSSAXLE-001	Hip 2-DOF Cross Axle	1
	LBPRT-TOPAXLE-001	Cross Axle Center	1
	LBPRT-TOPAXLE-002	Front-Back Axle Shaft	1
	LBPRT-TOPAXLE-003	Left/Right Axle Shafts	1
	LBPRT-TOPAXLE-004	Left/Right Axle Shafts	1
	HCDGH6-25	Misumi 25mm Pin	2
	HCDGH6-35	Misumi 35mm Pin	1
2	GEAKB1.5-32-15-B-15N	Misumi Spur Gear	1
3	LBPRT-KEY-002	5mm x 5mm Key	1
4	STWN15	Misumi Spring Clip	4



University of Waterloo - Adaptive Systems Laboratory

AUTHOR
Safwan Choudhury
E5 5129 (ext 31475)

TITLE:
Hip (2) Subassembly Reference

MATERIAL
Aluminum 6061

SIZE
A

ASSEMBLY NO.
LBSUB-HIP-002

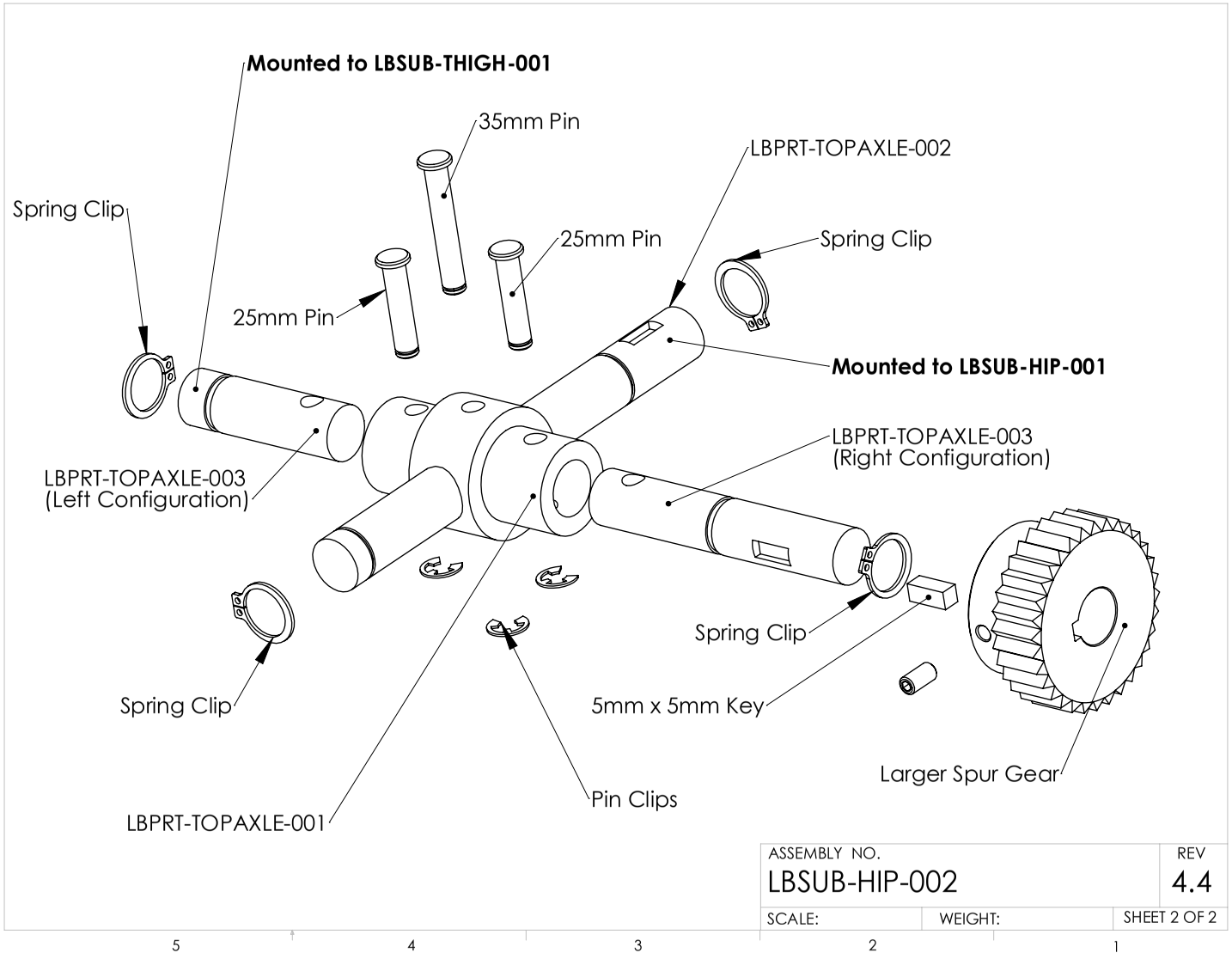
REV
4.4

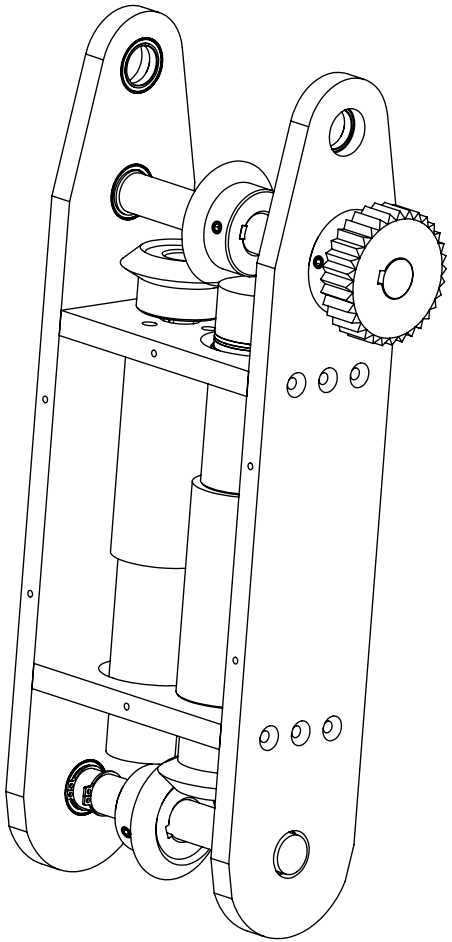
SCALE: 1:1

WEIGHT: 0.8kg

SHEET 1 OF 2

5 4 3 2 1


Mounted to LBSUB-THIGH-001
35mm Pin
LBPRT-TOPAXLE-002
Spring Clip
25mm Pin
Spring Clip
25mm Pin
Mounted to LBSUB-HIP-001
LBPRT-TOPAXLE-003
(Right Configuration)
LBPRT-TOPAXLE-003
(Left Configuration)
Spring Clip
Spring Clip
5mm x 5mm Key
Larger Spur Gear
Pin Clips
LBPRT-TOPAXLE-001
ASSEMBLY NO.
LBSUB-HIP-002
REV
4.4
SCALE:
WEIGHT:
SHEET 2 OF 2
5
4
3
2
1



ITEM	PART NUMBER	DESCRIPTION	QTY.
1	LBPRT-BRACE-001	Chassis Connecting Mounts	2
2	LBPRT-THIGH-001	Thigh Side Plates	2
3	LBPRT-KEY-002	5mm x 5mm Key	3
4	STWN15	Misumi Spring Clip	6
5	B6702ZZ	Misumi Ball Bearing	6
6	LBPRT-BUSHING-001	Keyed Motor Bushing	2
7	LBPRT-KEY-002	5mm x 5mm Key	2
8	KGEASK15-3030-15	Misumi Bevel Gear	4
9	LBPRT-DRILLROD-001	Top Shaft	1
10	GEAKB1.5-32-15-B-15N	Misumi Spur Gear	1
11	LBSUB-MOTOR-001	Larger Motor Assembly	2
	3257CR	Micromo DC Motor	1
	38A	Micromo Precision Gearhead	1
	HEDM5500	Micromo Encoder	1
12	LBPRT-DRILLROD-001	Bottom Shaft	1

University of Waterloo - Adaptive Systems Laboratory

AUTHOR
Safwan Choudhury
E5 5129 (ext 31475)

TITLE:
Thigh Subassembly Reference

MATERIAL
Aluminum 6061

SIZE
A

ASSEMBLY NO.
LBSUB-THIGH-001

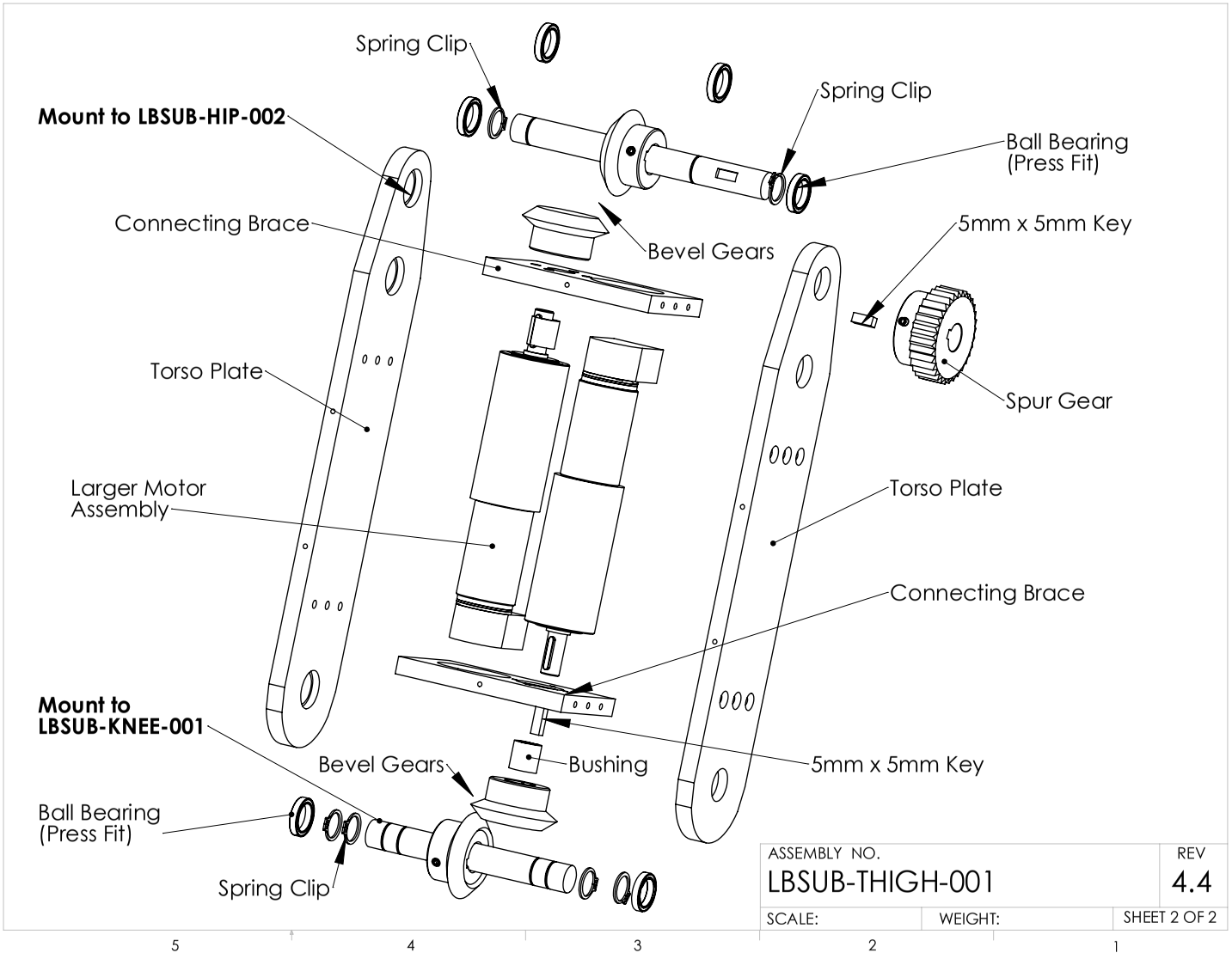
REV
4.4

SCALE: 1:3

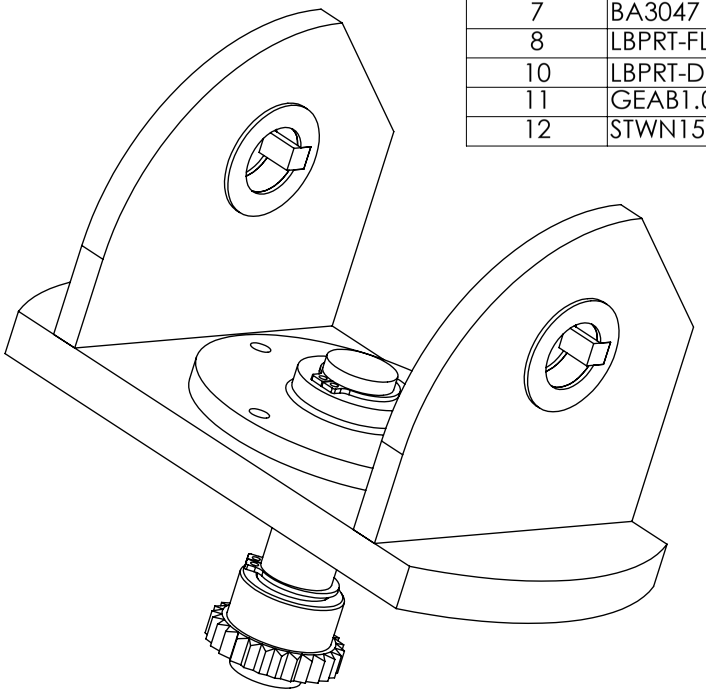
WEIGHT: 4kg

SHEET 1 OF 2

5 4 3 2 1


Spring Clip
Spring Clip
Mount to LBSUB-HIP-002
Ball Bearing
(Press Fit)
Connecting Brace
Bevel Gears
5mm x 5mm Key
Torso Plate
Spur Gear
Larger Motor
Assembly
Torso Plate
Connecting Brace
Mount to
LBSUB-KNEE-001
Bevel Gears
Bushing
5mm x 5mm Key
Ball Bearing
(Press Fit)
Spring Clip
ASSEMBLY NO.
LBSUB-THIGH-001
REV
4.4
SCALE:
WEIGHT:
SHEET 2 OF 2
5
4
3
2
1

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	LBPRT-KNEE-001	Knee Base Plate	1
2	LBPRT-KNEE-002	Knee Side Plates	2
3	LBPRT-KEY-002	5mm x 5mm Key	2
4	LBPRT-KNEE-003	Steel Reinforcement Washers	4
5	LBPRT-KEY-001	3mm x 3mm Key	1
6	LBPRT-WASHER-001	Machined Plastic Washer	1
7	BA3047	Misumi Needle-Roller Bearing	1
8	LBPRT-FLANGE-001	Machined Flange Component	1
10	LBPRT-DRILLROD-002	Joint Shaft	1
11	GEAB1.0-26-6-B-15	Misumi Spur Gear	1
12	STWN15	Misumi Spring Clip	3



University of Waterloo - Adaptive Systems Laboratory

AUTHOR
Safwan Choudhury
E5 5129 (ext 31475)

TITLE:
Knee Subassembly Reference

MATERIAL
Aluminum 6061

SIZE
A

ASSEMBLY NO.
LBSUB-KNEE-001

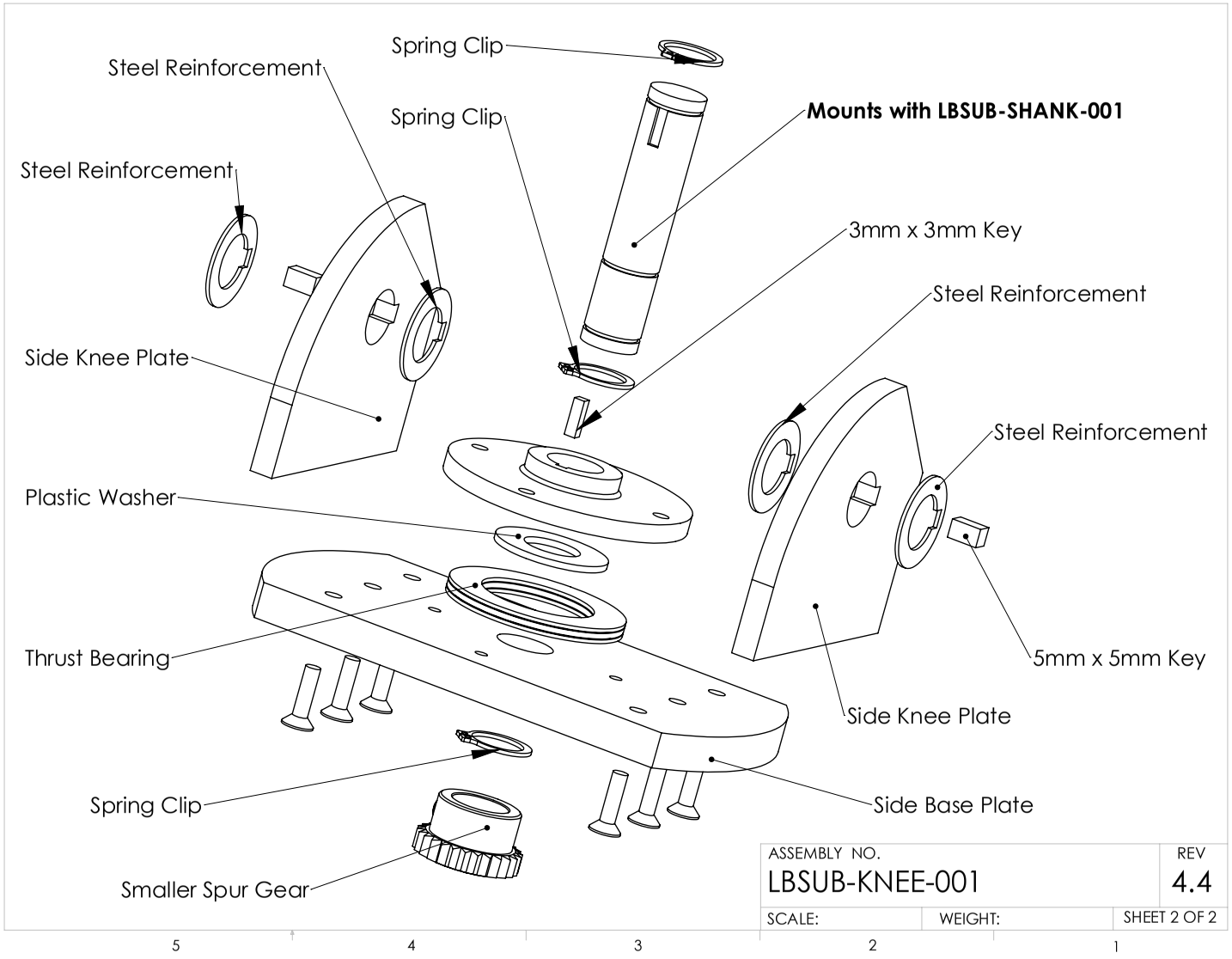
REV
4.4

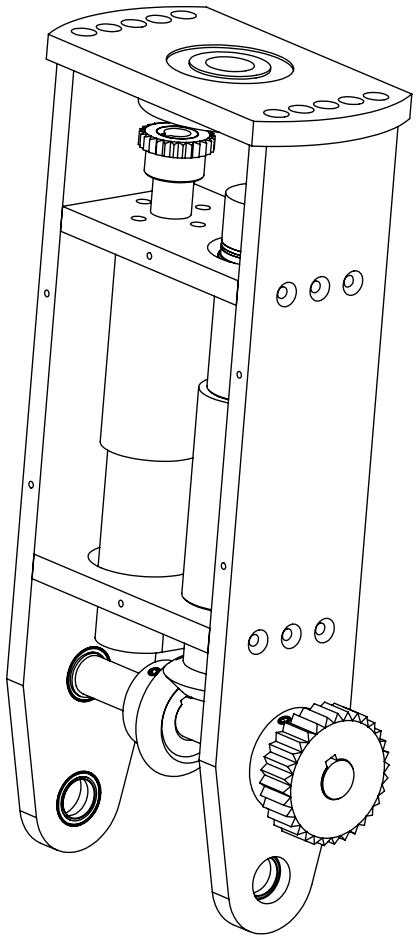
SCALE: 1:1.1

WEIGHT: 0.6kg

SHEET 1 OF 2

5 4 3 2 1


Spring Clip
Steel Reinforcement
Spring Clip
Mounts with LBSUB-SHANK-001
Steel Reinforcement
3mm x 3mm Key
Steel Reinforcement
Side Knee Plate
Steel Reinforcement
Plastic Washer
Thrust Bearing
5mm x 5mm Key
Side Knee Plate
Spring Clip
Side Base Plate
Smaller Spur Gear
ASSEMBLY NO.
LBSUB-KNEE-001
REV
4.4
SCALE:
WEIGHT:
SHEET 2 OF 2
5
4
3
2
1



ITEM	PART NUMBER	DESCRIPTION	QTY
1	LBPRT-BRACE-001	Chassis Connecting Mounts	2
2	LBPRT-SHANK-002	Shank Top Plate	1
3	LBPRT-SHANK-001	Shank Side Plates	2
4	LBPRT-WASHER-001	Machined Plastic Washer	1
5	LBPRT-BUSHING-003	Shank Shaft Adapter	1
6	BA3047	Misumi Needle-Roller Bearing	1
7	LBPRT-FLANGE-001	Machined Flange Component	1
8	GEAB1.0-26-6-B-15	Misumi Spur Gear	1
9	LBSUB-MOTOR-001	Larger Motor Assembly	2
10	B6702ZZ	Misumi Ball Bearing	4
11	LBPRT-KEY-002	5mm x 5mm Key	2
12	LBPRT-KEY-002	5mm x 5mm Key	1
13	LBPRT-BUSHING-001	Keyed Motor Bushing	1
14	STWN15	Misumi Spring Clip	2
15	GEAKB1.5-32-15-B-15N	Misumi Spur Gear	1
16	KGEASK15-3030-15	Misumi Bevel Gear	2
17	LBPRT-DRILLROD-001	Joint Shaft	1

University of Waterloo - Adaptive Systems Laboratory

AUTHOR
Safwan Choudhury
E5 5129 (ext 31475)

TITLE:
Shank Subassembly Reference

MATERIAL
Aluminum 6061

SIZE
A

ASSEMBLY NO.
LBSUB-SHANK-001

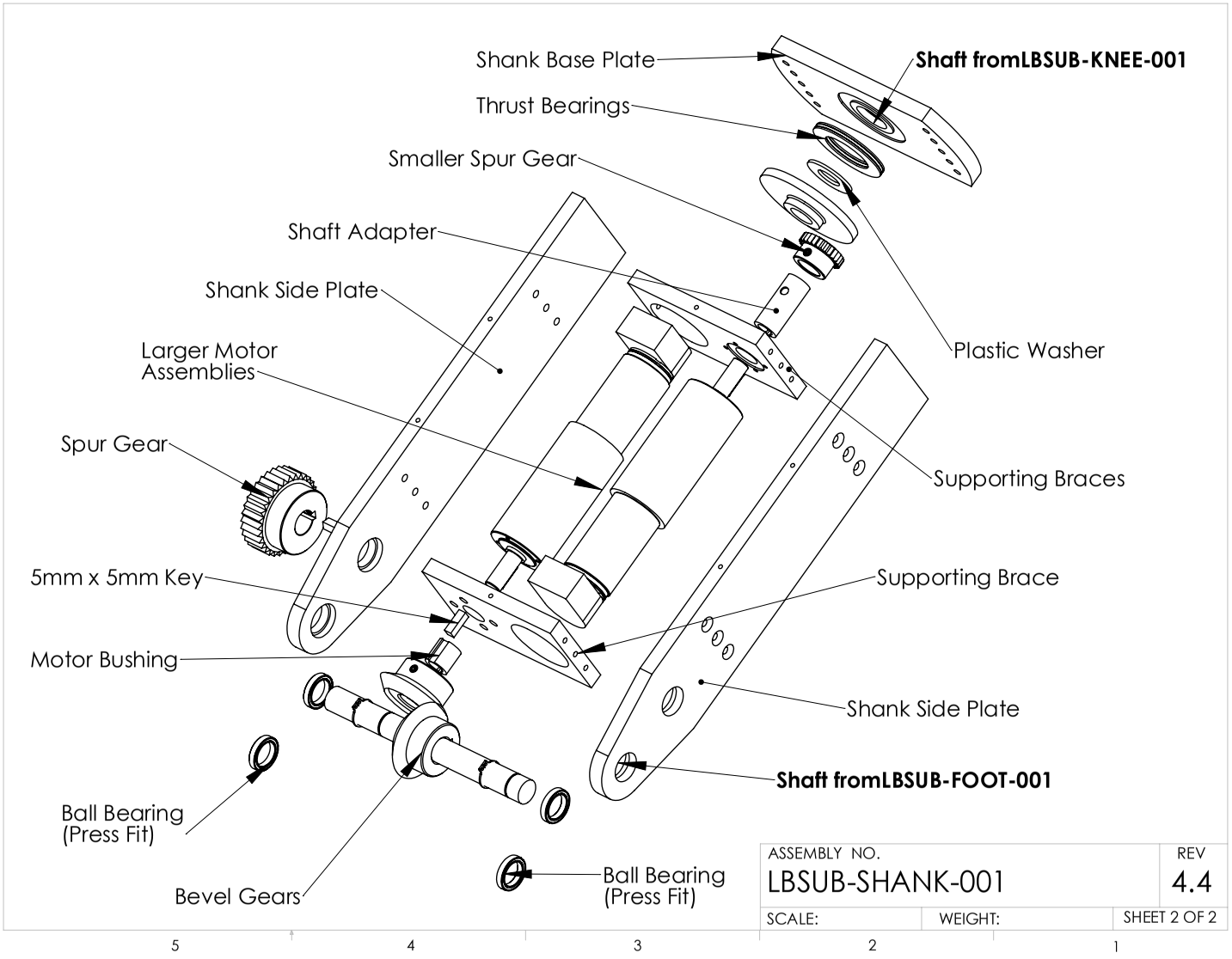
REV
4.4

SCALE:1:2

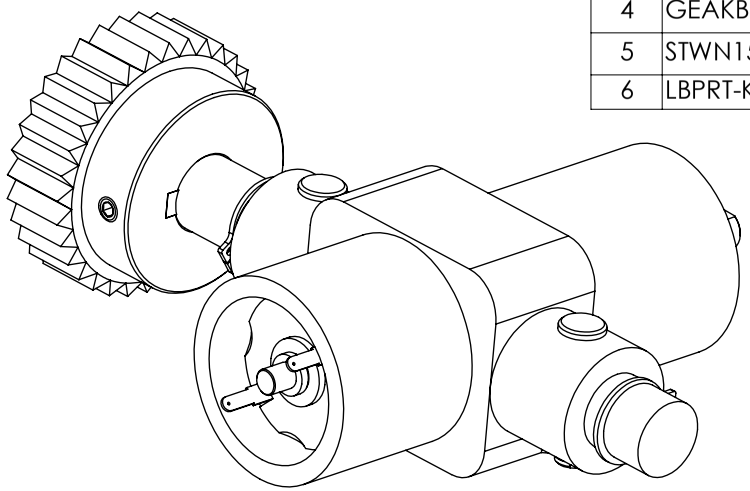
WEIGHT: 3.9kg

SHEET 1 OF 2

5 4 3 2 1


Shank Base Plate
Shaft fromLBSUB-KNEE-001
Thrust Bearings
Smaller Spur Gear
Shaft Adapter
Shank Side Plate
Plastic Washer
Larger Motor Assemblies
Spur Gear
Supporting Braces
5mm x 5mm Key
Supporting Brace
Motor Bushing
Shank Side Plate
Shaft fromLBSUB-FOOT-001
Ball Bearing (Press Fit)
Bevel Gears
Ball Bearing (Press Fit)
ASSEMBLY NO.
LBSUB-SHANK-001
REV
4.4
SCALE:
WEIGHT:
SHEET 2 OF 2
5
4
3
2
1

ITEM	PART NUMBER	DESCRIPTION	QTY.
1	LBSUB-CROSSAXLE-002	Foot 2-DOF Cross Axle	1
	LBPRT-BOTTOMAXLE-001	Cross Axle Center	1
	LBPRT-BOTTOMAXLE-002	Left/Right Shafts	1
	LBPRT-BOTTOMAXLE-002	Left/Right Shafts	1
	HCDGH6-25	Misumi 25mm Pin	2
2	LBSUB-MOTOR-004	Smaller Motor Assembly	1
	3242CR	Micromo 3242CR	1
	32A	Micromo 32A	1
4	GEAKB1.5-32-15-B-15N	Misumi Spur Gear	1
5	STWN15	Misumi Spring Clip	2
6	LBPRT-KEY-002	5mm x 5mm Key	1



University of Waterloo - Adaptive Systems Laboratory

AUTHOR
Safwan Choudhury
E5 5129 (ext 31475)

TITLE:
Foot (1) Subassembly Reference

MATERIAL
Aluminum 6061

SIZE
A

ASSEMBLY NO.
LBSUB-FOOT-001

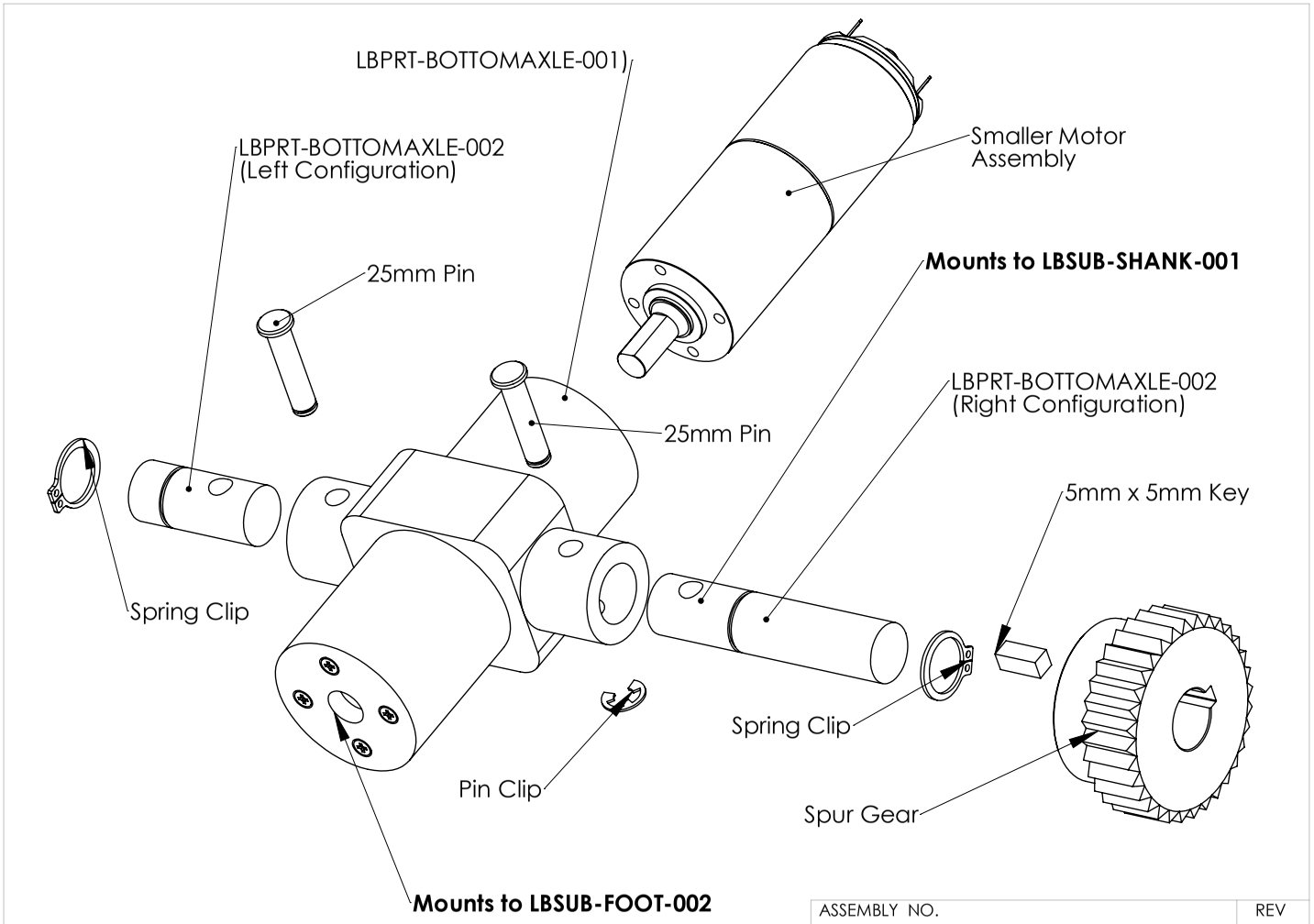
REV
4.4

SCALE: 1:1

WEIGHT: 0.9kg

SHEET 1 OF 2

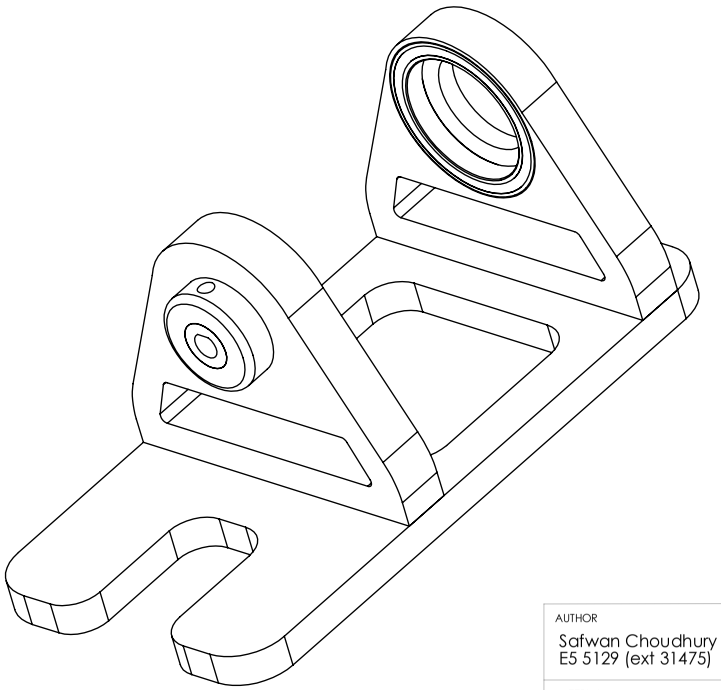
5 4 3 2 1



ASSEMBLY NO.		REV
LBSUB-FOOT-001		4.4
SCALE:	WEIGHT:	SHEET 2 OF 2

5 4 3 2 1

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	LBPRT-FOOT-001	Foot Front/Back Plate	1
2	LBPRT-FOOT-001	Foot Front/Back Plate	1
3	LBPRT-FOOT-002	Foot Base Plate	1
4	LBPRT-BUSHING-002	Joint Shaft	1
5	B6808ZZ	Misumi Ball Bearing B6808ZZ	2



University of Waterloo - Adaptive Systems Laboratory

AUTHOR
Safwan Choudhury
E5 5129 (ext 31475)

TITLE:
Foot (2) Subassembly Reference

MATERIAL
Aluminum 6061

SIZE
A

ASSEMBLY NO.
LBSUB-FOOT-002

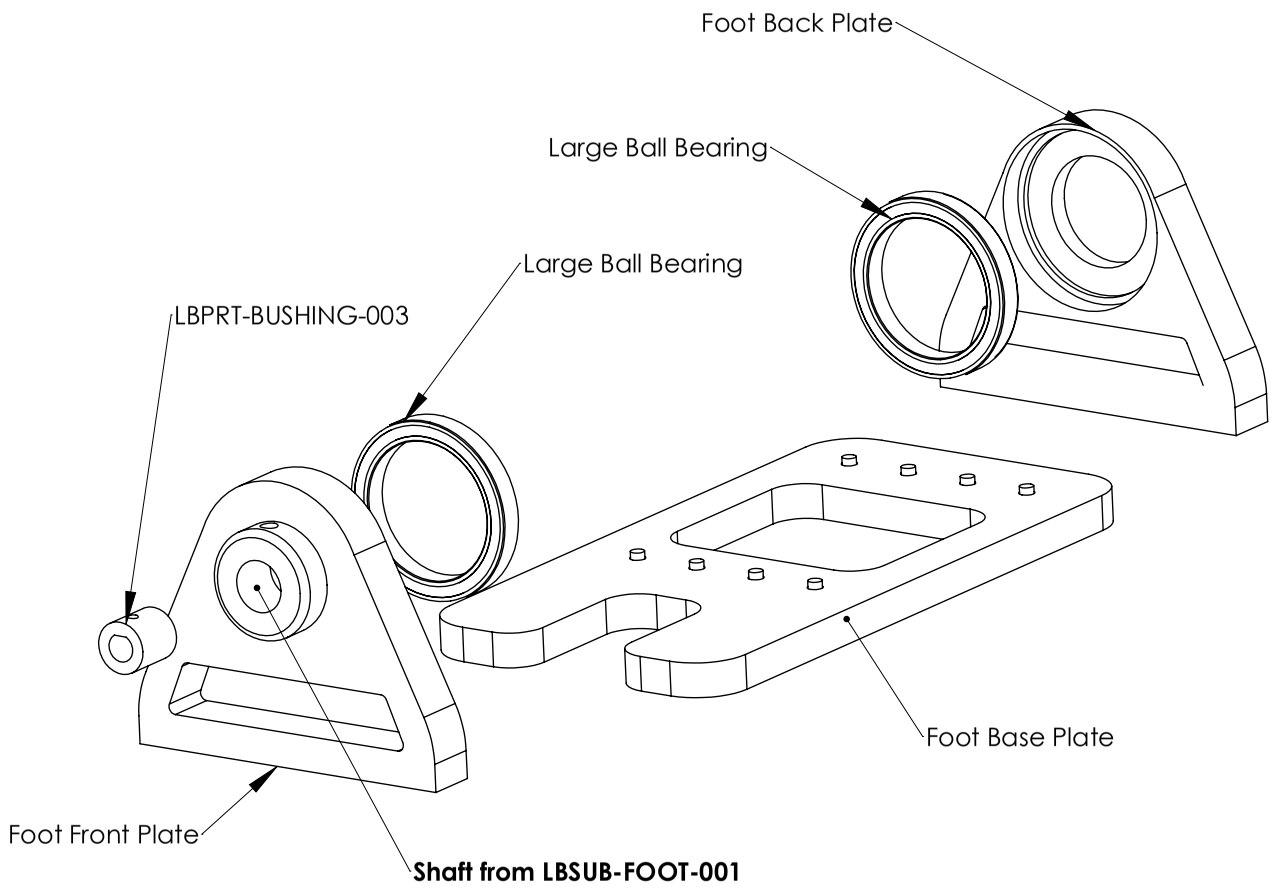
REV
4.4

SCALE: 1:1.5

WEIGHT: 0.7kg

SHEET 1 OF 2

5 4 3 2 1



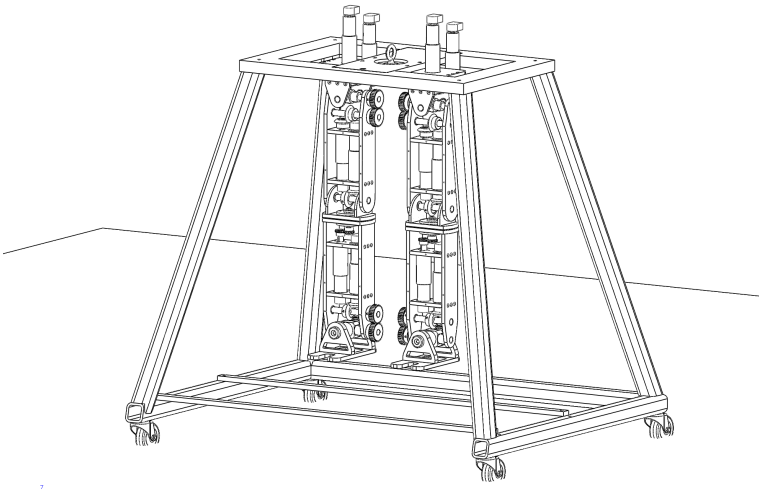
ASSEMBLY NO.		REV
LBSUB-FOOT-002		4.4
SCALE:	WEIGHT:	SHEET 2 OF 2

5 4 3 2 1

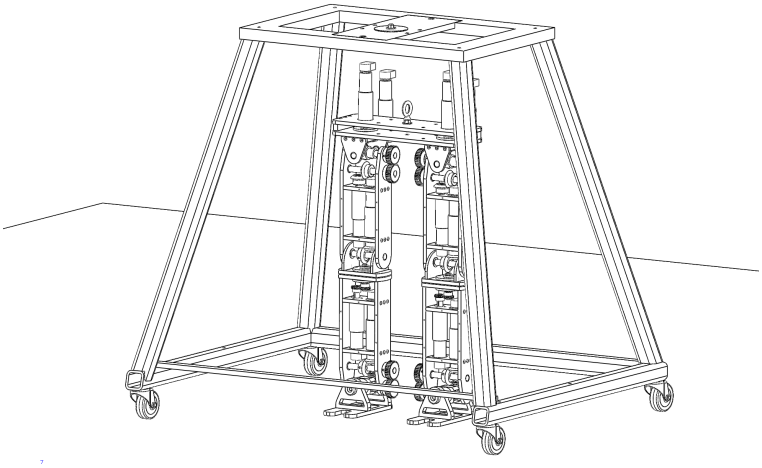
A.2 Walking Frame

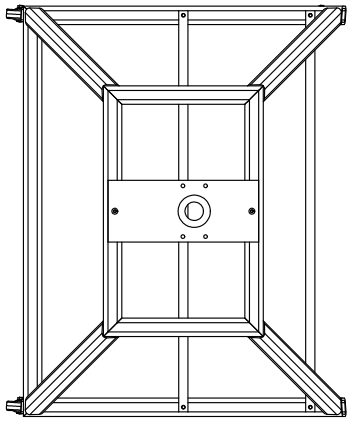
The supporting frame was designed for fixed base and (tethered) walking experiments.

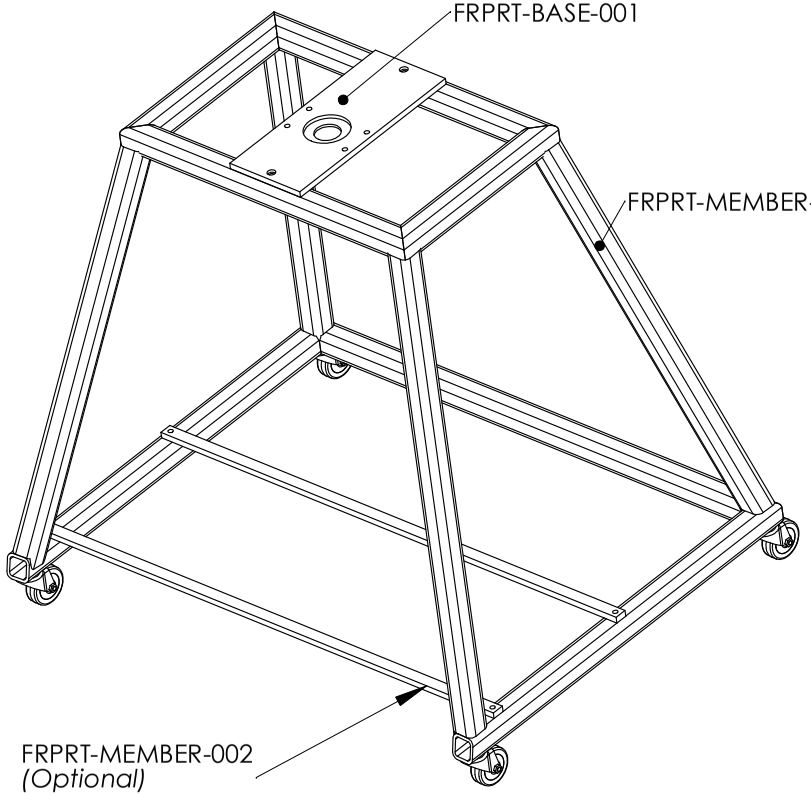
Fixed Configuration

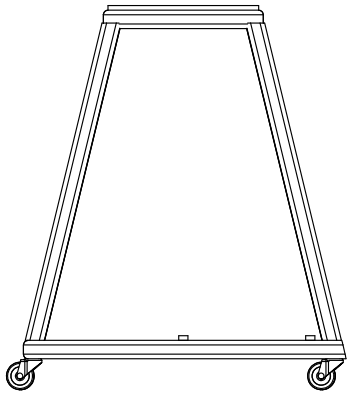


Walking Configuration





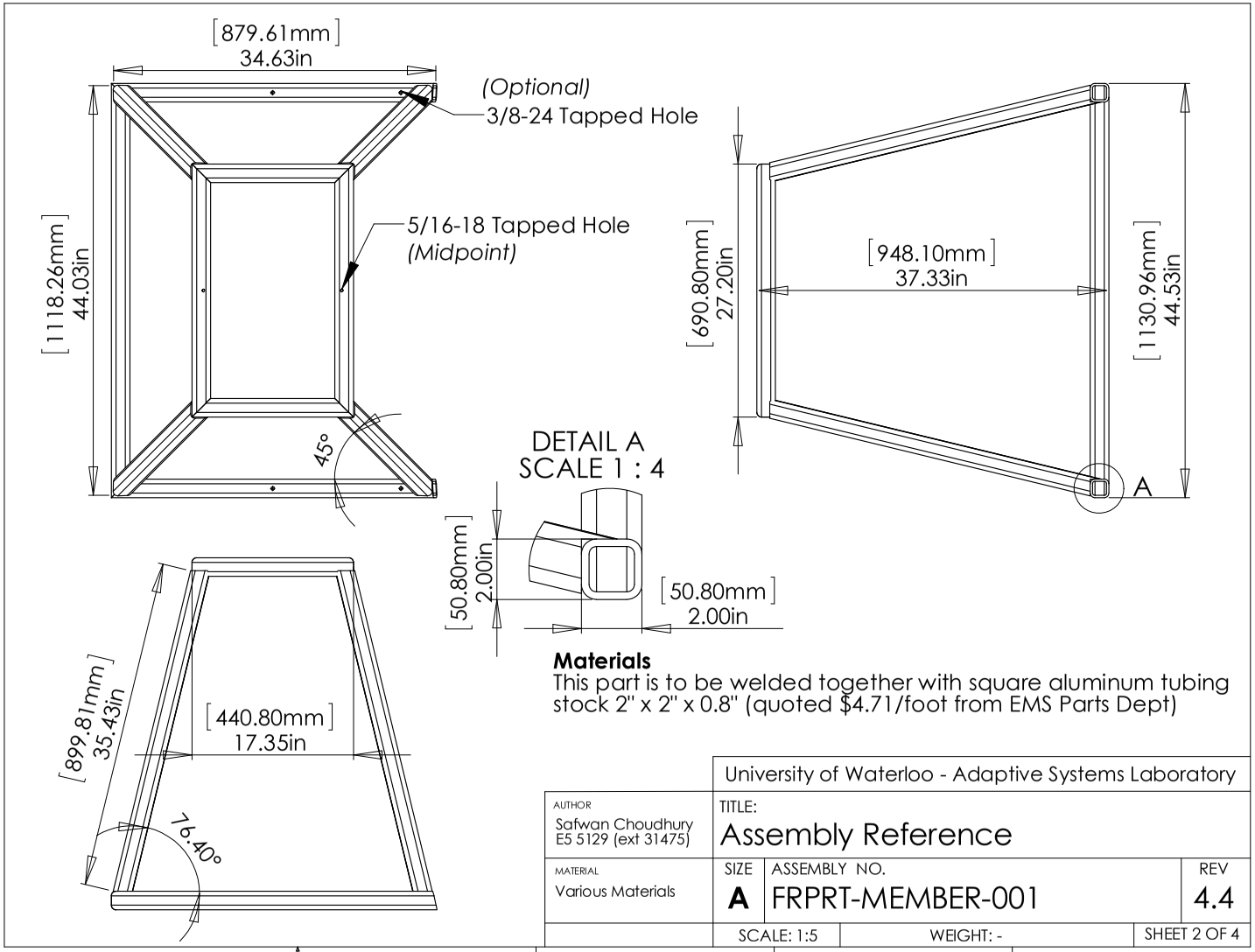




University of Waterloo - Adaptive Systems Laboratory

AUTHOR: Safwan Choudhury E5 5129 (ext 31475)	TITLE: Assembly Reference		
MATERIAL: Various Materials	SIZE: A	ASSEMBLY NO.: ASLHP-FRAME	REV: 4.4
	SCALE: 1:5	WEIGHT: -	SHEET 1 OF 4

5 4 3 2 1



Materials
This part is to be welded together with square aluminum tubing stock 2" x 2" x 0.8" (quoted $4.71/foot from EMS Parts Dept)

	University of Waterloo - Adaptive Systems Laboratory		
AUTHOR: Safwan Choudhury E5 5129 (ext 31475)	TITLE: Assembly Reference		
MATERIAL: Various Materials	SIZE: A	ASSEMBLY NO.: FRPRT-MEMBER-001	REV: 4.4
	SCALE: 1:5	WEIGHT: -	SHEET 2 OF 4

5 4 3 2 1